LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

510.84
I16r
no.667-672
cop.2

510.84
I16r
no.672
Copz

UIUCDCS-R-74-672

Math

COO-2383-0012

MULTI-DERIVATIVE NUMERICAL METHODS FOR THE SOLUTION OF STIFF ORDINARY DIFFERENTIAL EQUATIONS

BY

ROY LEONARD BROWN, JR.

August, 1974

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

THE LIBRARY OF THE
FEB 24 1975
UNIVERSITY OF ILLINOIS

UIUCDCS-R-74-672 COO-2383-0012

MULTI-DERIVATIVE NUMERICAL METHODS FOR THE SOLUTION OF STIFF ORDINARY DIFFERENTIAL EQUATIONS

BY

ROY LEONARD BROWN, JR.

August, 1974

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

* Supported in part by the Atomic Energy Commission under contracts US AEC AT(11-1)2383 and US AEC AT(11-1)1469 and submitted in partial fulfillment of the requirements of the Graduate College for the degree of Doctor of Philosophy in Computer Science.

Digitized by the Internet Archive
in 2013

http://archive.org/details/multiderivativen672brow

MULTI-DERIVATIVE NUMERICAL METHODS FOR THE SOLUTION OF STIFF ORDINARY DIFFERENTIAL EQUATIONS

Roy Leonard Brown, Jr., Ph.D.
Department of Computer Science
University of Illinois at Urbana-Champaign, 1974

Current research in the physical sciences and engineering has greatly enhanced the importance of computer programs for the numerical solution of systems of ordinary differential equations. Recent developments have concentrated on the solution of systems characterized as stiff for which the requirement of numerical stability can cause some integration methods to use a smaller stepsize than the control of the local truncation error requires.

The stability and convergence of linear multi-step multi-derivative formulas have been investigated since it has been shown that linear multi-step formulas of order greater than two cannot be A-stable, a useful attribute for formulas used to integrate stiff systems. It has been shown that when a stable and consistent linear k-step s-derivative formula implemented in an iterated corrector method is applied to a system satisfying certain continuity requirements, the solution is convergent.

A set of 11 multi-derivative multi-step formulas, all A-stable and of orders 1 through 11, have been found and implemented in a variable stepsize, variable order method as a FORTRAN program called D4. Tests of D4 on a set of stiff problems show it compares favorably with another stiff integrator.

ACKNOWLEDGMENT

I wish to express my gratitude to my advisor, Professor C. W. Gear, for his sure guidance, constructive criticism, and valuable support during the development of this work. D. S. Watanabe, R. D. Skeel, and the other members of the weekly numerical analysis seminar contributed much by listening to several presentations on the related research. The ALTRAN compiler and its software support were made available by the Math and Numerical Services Unit of the Computing Services Office, University of Illinois. A. C. Norman provided the TAYLOR compiler and execution supervisor, as well as valuable suggestions. Computing time was provided by the U. S. Atomic Energy Commission under grants 1469 and 2383.

Special thanks are due Pam Farr for her efficient typing of the draft and final versions of the manuscript.

This work is dedicated to the memory of the mathematician C. L. Dodgson (1832-1898).

TABLE OF CONTENTS

	Page
1. Introduction: Stiffness and Stability	1
1.1 Stiff Equations	2
1.2 A-Stability	4
2. Multi-Derivative Formulas	9
2.1 A Survey of Higher Derivative Methods	9
2.2 Some Formulas Well-Suited to Stiff Problems	12
2.3 Convergence and Error Estimation for Multi-derivative Formulas	18
3. Implementation	35
3.1 The Iterated Corrector Method	35
3.2 Stepsize and Order Changing	41
3.3 Program Description	44
3.4 Properties of Stepsize and Order Changing Methods	45
3.5 Higher Derivative Calculation	53
3.6 Numerical Tests	56
4. Alternatives and Conclusions	59
4.1 Matrix-Valued Coefficient Methods	59
4.2 Conclusions	66
LIST OF REFERENCES	69
APPENDICES	
I D4 FOR THE INTEGRATION OF STIFF SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS	72
II TEST PROBLEMS FOR COMPARISON OF D4 WITH DIFSUB	86

1. Introduction: Stiffness and Stability

Current research in the physical sciences and engineering has greatly enhanced the importance of computer programs for the numerical solution of systems of ordinary differential equations. The usage of both special purpose and general programs has increased, and, in particular, the availability of methods for solving systems characterized as stiff has led to an increase in attempts to solve such systems. For examples of both stiff problems and current methods used to solve them, see Bjurel et al [3]. This work will present a method that should solve a large subclass of all stiff systems; however, the amount and type of information it requires, which includes the second or higher derivatives of a first order system, place it as a special purpose program suitable for certain problems which more general computer packages are unable to solve efficiently.

The method will be developed in several steps. First, the underlying formulas on which the method is based will be derived and shown to meet certain previously defined desirable criteria. Second, the program package design will be considered, with justification offered for the design choices made. Third, the results of tests comparing the program with a similar package having a similar program organization will be presented and discussed.

The desirable criteria of A-stability, convergence, and stability at infinity are presented in Chapter 1 following a discussion of stiffness in the next section.

In Chapter 2, conditions necessary for A-stability and sufficient for stability at infinity are derived and used to search out a set of formulas that satisfy these desirable criteria. A theorem on the convergence of these formulas is proved, and an asymptotic error estimate is derived which provides formulas used by the method to calculate stepsize and select order. Some observations are made about a recent conjecture of Genin [24] concerning the maximum order attainable by multi-derivative A-stable formulas. Chapter 3 presents some details of the program implementation with notes on program usage. The effect of changing stepsize and order is considered. Test results are presented. Chapter 4 presents a more easily used matrix-valued coefficient formula with stability properties similar to those of A-stable multi-derivative methods. Concluding observations on the usefulness of both multi-derivative methods and matrix-valued coefficient methods are made.

1.1 Stiff Equations

The solution of problems in such areas as control theory, chemical kinetics, circuit theory, and nuclear reactor kinetics requires modeling of physical systems by large systems of ordinary differential equations. If only Lipschitz problems, as defined below, are considered, then a unique solution exists. For a standard proof, see Coppel [9 : p. 12].

Definition 1.1

A Lipschitz problem is the initial value problem

$$\underline{y}' = \underline{f}(\underline{y}, t), \qquad (1.1)$$
$$\underline{y}(t_0) = \underline{y}_0,$$

such that $\underline{f}(\underline{y},t)$ is continuous in t and satisfies the Lipschitz condition

$$|| \underline{f}(\underline{y}_1, t) - \underline{f}(\underline{y}_2, t) || \leq L || \underline{y}_1 - \underline{y}_2 ||$$

for some positive L in the region $t_0 \leq t \leq t_f$, $|| \underline{y} || < \infty$.

A Lipschitz problem is considered stiff if its numerical solution has certain stability properties that make the computation difficult. For example, consider the system

$$\underline{y}' = A \underline{y}, \qquad (1.2)$$
$$\underline{y}(t_0) = \underline{y}_0,$$

where A is a real n x n matrix with its eigenvalues $\{\lambda_i\}$ satisfying $\text{Re}(\lambda_i) < 0$, at least two solutions are not identically zero, and

$$\sigma = \max_{i,j} \frac{| \text{Re}(\lambda_i) |}{|\lambda_j|}$$

is large in the sense that it is greater than some threshold value σ_0. Ehle [15] classifies a range of stiff problems with stiffness ratios σ varying from mildly stiff for $10 \leq \sigma \leq 100$ to extremely stiff for $\sigma > 10^6$.

Equations (1.2) are stable since the true solution

$$\underline{y}(t) = e^{(t-t_0)A} \underline{y}_0 \qquad (1.3)$$

approaches zero exponentially as t increases. In fact, the non-zero component corresponding to the largest $| \text{Re}(\lambda_i) |$ becomes small enough that its contribution to the truncation error of the numerical solution becomes negligible before any other component.

Yet for standard techniques such as the Adams method (see Bashforth and Adams [2] and Moulton [34]), and explicit Runge-Kutta formulas (see Ralston [37: 199]), absolute stability, to be considered shortly, requires that

$$\max_i | h\lambda_i | < C \qquad (1.4)$$

where h is the stepsize in the independent variable and C is some function dependent on the numerical method. Therefore, a method suitable for stiff equations is one which places no upper limit on $| h\lambda_i |$ where λ_i is an eigenvalue of the locally linearized system $\underline{y}' = \underline{f}_{\underline{y}}\, \underline{y}$, for $Re(\lambda_i) \leq 0$, other than restrictions based on controlling the error of the computation. Here $\underline{f}_{\underline{y}} = \frac{d\underline{f}}{d\underline{y}}$, the Jacobian matrix of $\underline{f}$ evaluated at $\underline{y}$ and t.

Ehle [15] has proposed that a system be called stiff only with respect to a method. Thus, a problem p is step-wise-stiff with respect to method m_1 if the average stepsize required in solving p with m_1 is smaller than that used in solving p with some other method m_2, other parameters being equal. Thus, finding appropriate stiff methods is equivalent to a search for maximal elements of the set $\{m_i\}$ with respect to step-wise-stiffness for all p in some class of problems P.

Further readings in stiff systems can be found in Bjurel *et al* [3], Ehle [15], and in the works listed in Appendix II of Enright [16].

1.2 A-stability

Stiff equations were first discussed as a problem in numerical analysis by Curtiss and Hirschfelder [10], who encountered the problem in chemical kinetic research. Dahlquist [11, 12] first provided a general treatment of the stability of multistep formulas, defined below.

Definition 1.2

A k-step multistep formula for problem (1.1) has the form

$$0 = \sum_{i=0}^{k} \alpha_i^{(0)}\, y_{n-i} + \alpha_i^{(1)}\, f(y_{n-i}, t_{n-i}) \qquad (1.5)$$

where y_n approximates $y(t_n)$ and $t_n = t_0 + nh$.

Dahlquist [13] subsequently investigated the special properties

necessary for stiff multistep methods. Gear [18: 116-132] expanded this work to include his multivalue methods which treat directly the elements of the vector used to store past information. For multistep methods this vector is

$$\underline{y}_n = (y_n, \ldots, y_{n-k}, hy'_n, \ldots, hy'_{n-k})^T,$$

where T is the transpose operator. In a multivalue method, this vector may be transformed to store differences, Taylor series, etc. Multistep methods are a subset of multivalue methods, so the latter will be discussed here.

The main construct in the theoretical treatment is the stability region of the test equation

$$y' = \lambda y, \qquad (1.6)$$
$$y(t_0) \neq 0.$$

Definition 1.2

The stability region R associated with a multivalue formula is the set $R = \{h\lambda$: the formula applied to (1.6) with constant stepsize $h > 0$ produces a sequence $\{y_n\}$ satisfying $y_n \to 0\}$.

When a method is applied to a system of equations $\underline{y}' = A\underline{y}$, it is absolutely stable for values of $| h\lambda_i | \in R$ where λ_i are the eigenvalues of the Jacobian matrix A. The most desirable property for a multivalue formula suitable to stiff problems is that $| h\lambda_i |$ never be restricted by the requirement of absolute stability for $Re(\lambda_i) < 0$. This property is defined as follows.

Definition 1.3

A multistep formula is A-stable if its associated stability region contains the open left complex half plane.

Dahlquist [13] also proved that the highest order attainable by an A-stable multistep method (1.5) is two i.e. the local truncation error

$$L_h(y(t_n)) = \sum_{j=0}^{k} \alpha_j^{(0)} y(t_{n-i}) + \alpha_j^{(1)} f(y(t_{n-j}), t_{n-j}) = O(h^3) \quad (1.7)$$

This limitation also applies to multivalue methods (see Gear [21]). To avoid this problem, Widlund [41] and Gear [18] relaxed the restriction of A-stability in an attempt to find higher order multivalue formulas suitable for stiff problems.

Definition 1.4 (Widlund)

A formula is A(α)-stable if its stability region R contains the wedge $S(\alpha) = \{z: \mid \arg(-z) \mid < \alpha\}$. (See Figure 1.)

Definition 1.5 (Gear)

A formula is stiffly-stable if its stability region contains a region of the form $R_1 \cup R_2$ where

$$R_1 = \{Re(z) \leq D < 0\}$$

$$R_2 = \{|Im(z)| < \theta, D \leq Re(z) \leq 0\}.$$

(See Figure 2.)

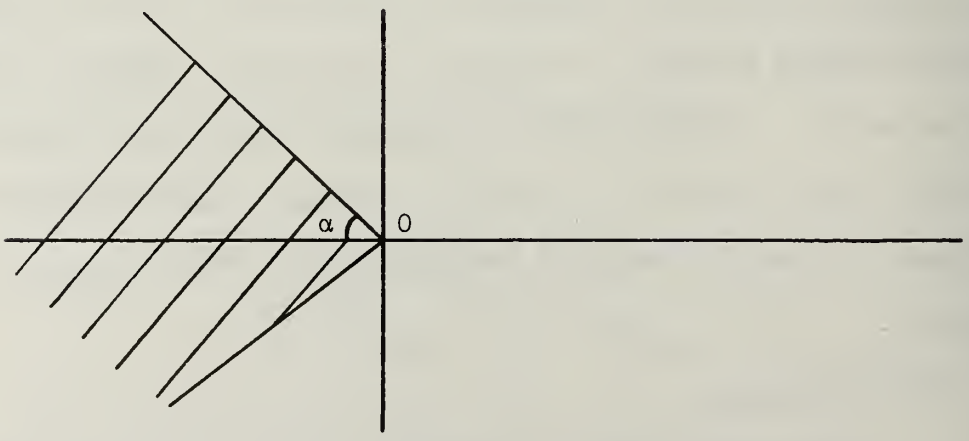

Figure 1. Wedge-shaped region $S(\alpha)$ associated with A(α)-stability

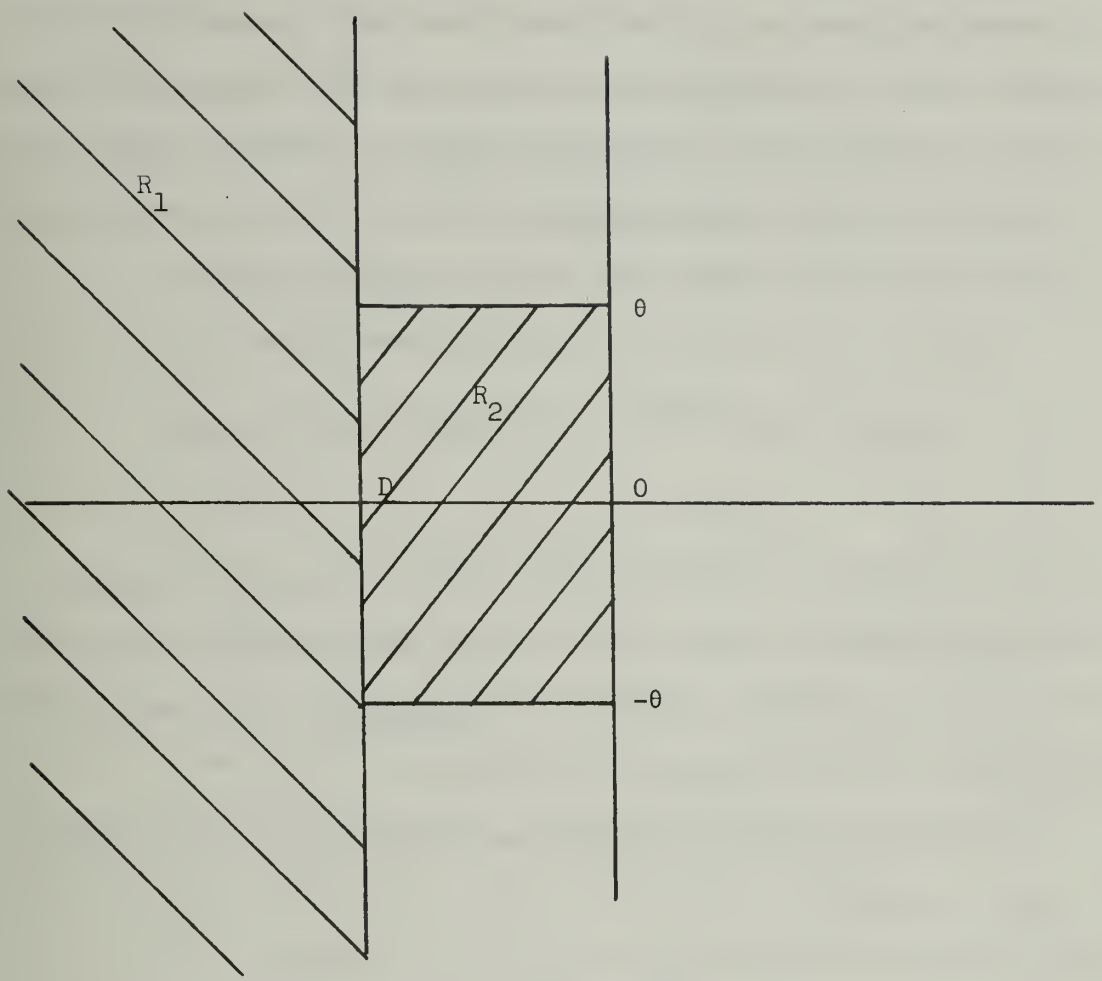

Figure 2. Region $R_1 \cup R_2$ associated with stiff stability

Stiffly stable formulas can be seen to be A(α)-stable for $\alpha = \tan^{-1}|\theta/D|$. Gear [20,21] used stiffly-stable backward differentiation formulas of order up to six and implemented them in a program DIFSUB.

Some additional concepts concerning multivalue formulas are needed. For a multistep formula of the form (1.5) define the polynomials

$$\rho(\zeta) = \sum_{i=0}^{k} \alpha_i^{(0)} \zeta^{k-i} , \qquad (1.8)$$

$$\sigma(\zeta) = \sum_{i=0}^{k} \alpha_i^{(1)} \zeta^{k-i} .$$

Absolute stability requires the zeros of $\rho(\zeta) + h\lambda\sigma(\zeta)$ to be less than or equal to 1 in norm. If all the zeros of $\rho(\zeta)$ are inside the unit circle except for the simple principal root, $\zeta_1 = 1$, then the formula is said to satisfy the strict root condition. A method based on such a formula is called strongly stable. If other zeros on the unit circle exist but are still simple, the method is weakly stable.

If the discretization error of the formula is

$$L_h(y(t_n)) = \sum_{i=0}^{k} \alpha_i^{(0)}\, y(t_{n-i}) + h\alpha_i^{(1)}\, f(y(t_{n-i}),t_{n-i}) \qquad (1.9)$$

$$= O(h^{p+1})$$

then the formula has order p. If $p \geq 1$, the formula is consistent. Consistency and the strict root condition are sufficient conditions for a fixed step multistep formula to be convergent i.e. y_n converges to $y(t_n)$ as $n \to \infty$ for stepsize $h = (t_f - t_0)/n$.

One final requirement precludes methods with the undesirable property that

$$\lim_{n \to \infty} \left| \frac{y_n}{y_{n-1}} \right| \to 1$$

as $h\lambda \to -\infty$, where y_n is the calculated solution at $t_0 + nh$ to (1.6).

Definition 1.6

A formula is stable at infinity if there exists a real $W < 0$ such that

$$\sup_{\mathrm{Re}(h\lambda) < W} \lim_{n \to \infty} \left| \frac{y_n}{y_{n-1}} \right| < 1, \qquad (1.10)$$

where y_n is the multivalue method solution to the test equation at $t_0 + nh$ for fixed h.

2. Multi-Derivative Formulas

There are many examples in the literature of formulas that require derivatives higher than the first to solve first order systems of ordinary differential equations. Some authors provide justifications for requiring higher derivatives by demonstrating techniques for calculating them; others do not. Throughout the remainder of this work, a k-step s-derivative multivalue method will be defined by

$$0 = \sum_{i=0}^{k} (\alpha_i^{(0)} \underline{y}_{n-i} + \sum_{j=1}^{s} \alpha_i^{(j)} h^j \underline{f}^{(j-1)} (\underline{y}_{n-i}, t_{n-i})), \qquad (2.1)$$

for some $\alpha_i^{(s)} \neq 0$.

In section 2.1, some examples are presented to motivate the desirable criteria of A-stability and ε-stability (related to stability at infinity). Section 2.2 presents some conditions related to A-stability and ε-stability. A method is described for graphing the stability regions of multiderivative formulas, and a set of A-stable and stable-at-infinity formulas are presented that were found by using this graphical method.

Section 2.3 contains a convergence theorem and an asymptotic error theorem, as well as comments on a constructive demonstration of a conjecture about the maximum order of an A-stable multiderivative method. The formulas presented in Section 2.2 fall within the bounds set by this conjecture.

2.1 A Survey of Higher Derivative Methods

Ehle [14] analyzes Obrechkoff's one-step formulas

$$0 = -y_n + y_{n-1} + \sum_{j=1}^{s} h^j (\alpha_0^{(j)} y_n + \alpha_1^{(j)} y_{n-1}) \qquad (2.2)$$

which can be of order 2s and A-stable but are then not stable at infinity. Order 2s is achieved when $\alpha_0^j = (-1)^{j+1} \alpha_1^j$ are the coefficients of the jth diagonal Padé approximation to e^t (see Graves-Morris [25]). Other one-step

methods, all based on Padé approximations, are derived in Blue & Gummel [4], Cavendish et al [8], and Varga [40]. Since these methods are of order at most 2s a great deal of work must be done in comparison with other multi-derivative methods, since they attain lower order than the conjectured maximum order of an A-stable multi-derivative method, as presented in Section 2.3.

Liniger and Willoughby [32] consider the two parameter class of formulas:

$$0 = -y_n + y_{n-1} + \frac{h}{2}[(1-a)\, y'_{n-1} + (1+a)y'_n] + \frac{h^2}{4}[(b-a)\, y''_{n-1} - (b+a)\, y''_n]. \tag{2.3}$$

If $\frac{1}{3} \leq a + b \leq 2$ and $0 \leq b - a \leq \frac{1}{3}$, the formula is A-stable. By picking a and b appropriately the numerical solution can be made to follow certain components of the linearized problem

$$y' = \underline{f}_{\underline{y}}\, \underline{y}.$$

Enright [16] considers formulas

$$0 = -y_n + y_{n-1} + \sum_{i=0}^{k} \alpha_i^{(1)} f(y_{n-i}, t_{n-i}) + \alpha_0^{(2)} f'(y_n, t_n). \tag{2.4}$$

This provides formulas that are A-stable through order 4 and stiffly-stable through order 9. By requiring the equation to be autonomous of the form

$$\underline{y}' = \underline{f}(\underline{y}) \tag{2.5}$$
$$\underline{y}(t_0) = \underline{y}_0,$$

(easily accomplished by adding a new variable defined by $T' = 1$, $T(t_0) = t_0$, to the system), the second derivative can be evaluated by

$$\underline{f}'(\underline{y}_n) = \underline{f}_{\underline{y}} \underline{f}(\underline{y}_n). \tag{2.6}$$

Since most methods for stiff problems require a Newton type corrector iteration to provide rapid convergence for large stepsizes, some method for finding f_y, the Jacobian matrix, is usually available. However, an approximation to the Jacobian is usually sufficient. Therefore, many programs do not evaluate f_y exactly. Further, the procedure is time consuming and so a particular computed value is often carried over several steps. To require exact evaluation of f_y at every corrector iteration is perhaps excessive. See Brown [5] for similar second derivative formulas giving higher order A-stable methods.

Jeltsch [30] has developed techniques for studying multi-derivative methods. He calls

$$\rho(\zeta) + \sum_{j=1}^{s} \mu^j \sigma_j(\zeta) = 0, \qquad (2.7)$$

$$\mu = h\lambda,$$

the characteristic equation of a multivalue multi-derivative method (2.1), for $\rho(\xi)$ as in 1.8, and

$$\sigma_j(\zeta) = \sum_{i=0}^{k} \alpha_i^{(j)} \zeta^{k-i}. \qquad (2.8)$$

These equations are used to define the concept of ε-stability.

Definition 2.1

If the roots $\zeta_k(h\lambda)$ of the characteristic equation corresponding to the multivalue method satisfy for some finite k

$$|h\lambda|^{\eta} \max_i \{|\xi_i(h\lambda)|\} < k$$

for all $\eta \leq \varepsilon$ as $h\lambda \to -\infty$, then the method is ε-stable.

A sufficient but not necessary condition for a method to be stable at infinity is that the method be ε-stable for $\varepsilon > 0$. In fact,

$$\lim_{n \to \infty} \left| \frac{y_n}{y_{n-1}} \right| \to 0$$

as $h\lambda \to -\infty$ if a method is ε-stable for $\varepsilon > 0$.

Jeltsch shows that ε may be calculated geometrically from the Puiseux diagram corresponding to the method.

Definition 2.2

The Puiseux diagram for a method (2.1) is the 2-dimensional point set

$$P = \{(i,j) : \alpha_{k-i}^{(j)} \neq 0\}.$$

Puiseux diagrams for methods of Gear, Enright, and Jeltsch are shown in Figure 3. Jeltsch shows that ε is the smallest slope of all straight lines having a finite slope and passing through $\left(k, \max_{\alpha_0^{(j)} \neq 0} (j)\right)$ and any other point of P.

2.2 Some Formulas Well-Suited to Stiff Problems

A set of formulas that will solve any linear and most non-linear stiff problems should satisfy several criteria:

1) stability at infinity;
2) A-stability;
3) convergence with as high an order as possible to ensure as large a stepsize as possible and thus reduce the global error.

With this in mind the following definitions and lemmas are useful.

Definition 2.3

A multi-derivative multivalue formula (2.1) is implicit if at least one of $\alpha_0^{(j)} \neq 0$, $j \geq 1$.

Definition 2.4

A multi-derivative formula (2.1) is maximally implicit if

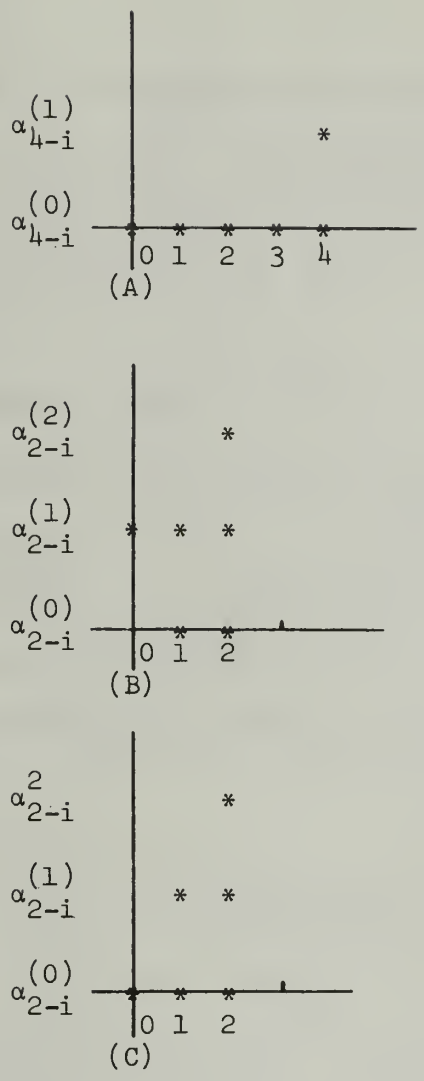

Figure 3. Puiseux diagrams of 4th order methods due to Gear (A), Enright (B), and Jeltsch (C).

$\alpha_0^{(s)} \neq 0.$

Lemma 2.1

If the stability region of a multi-derivative formula contains $h\lambda = -\infty$, then the formula is maximally implicit.

Proof

Consider the characteristic equation

$$0 = P(\zeta) = \rho(\zeta) + \sum_{j=1}^{s} \mu^j \sigma_j(\zeta) \tag{2.9}$$

and suppose $\alpha_0^{(s)} = 0$. Let the roots of $P(\zeta)$ be $\zeta_1(\mu),\ldots,\zeta_k(\mu)$. $P(\zeta)$ can be expressed as

$$P(\zeta) = \sum_{j=0}^{s-1} \alpha_0^{(j)} \mu^j \prod_{i=1}^{k} (\zeta - \zeta_i(\mu)). \tag{2.10}$$

It will be shown that requiring $\lim_{\mu \to \infty} \zeta_i(\mu)$ to be bounded leads to a contradiction. Consider the terms in $\zeta^{k-\ell}$ in each of the expansions (2.10) and (2.9). Setting them equal gives

$$\sum_{j=1}^{s-1} \alpha_0^{(j)} \mu^j \sum_{i_1<i_2<\ldots i_\ell} [-\zeta_{i_1}(\mu)][-\zeta_{i_2}(\mu)]\ldots[-\zeta_{i_\ell}(\mu)] \tag{2.11}$$

$$= \sum_{j=0}^{s} \alpha_\ell^{(j)} \mu^j.$$

Divide by μ^s and let $\mu \to -\infty$, obtaining

$$\alpha_\ell^{(s)} = 0.$$

Thus if $\alpha_0^{(s)} = 0$, then $\alpha_\ell^{(s)} = 0$, $\ell = 1,2,\ldots,k$, and the method cannot be an s-derivative method if $h\lambda = -\infty$ is in its stability region. Therefore for a method's stability region to contain $h\lambda = -\infty$, it must be maximally implicit. Q.E.D.

Since an A-stable formula must have $-\infty$ in its stability region, the requirement of maximal implicitness is a necessary condition for A-stability.

Lemma 2.2

If a method is maximally implicit and $\alpha_i^{(s)} = 0$ for $i \neq 0$, then it is ε-stable with $\varepsilon > 0$.

Proof

It is evident from the form of the Puiseux diagram, see Figure 3, that $\varepsilon > 0$ if $\alpha_0^{(s)} \neq 0$ and $\alpha_i^{(s)} = 0$ if $i \neq 0$. Q.E.D.

With this information, several restrictions are imposed on the formulas which will be considered for inclusion in a multivalue method of form (2.1). First, choose $\alpha_i^{(s)} = 0$ for $i > 0$, and $\alpha_0^{(s)} \neq 0$. Since $\alpha_0^{(s)} \neq 0$ requires that the s-th derivative of y_n be calculated, it will probably be worthwhile to calculate the 1-st through (s-1)-th derivatives as well, so $\alpha_0^{(j)} \neq 0$, $j = 1,\ldots,s$. A further heuristic limitation involves the belief that the most recent information, i.e. closest with respect to the independent variable to t_n, is the most useful. Thus, if a method is to use three values for y'', then $\alpha_0^{(2)}$, $\alpha_1^{(2)}$, $\alpha_2^{(2)}$ will be non-zero.

A program was written to graph the stability region boundary of a method described by $(s,\underline{m})$ where s is the highest derivative used, and $\underline{m}_j$ is the number of (most recent) evaluations of $y^{(j)}$, $j = 0,\ldots,s$. This program computes the coefficients $\alpha_i^{(j)}$ of the highest order method possible for a value $(s,\underline{m})$, and then plots the locus of μ for which the characteristic equation (2.7) is satisfied for any $\zeta = e^{2\pi i\theta}$, $0 \leq \theta \leq 1$. The following set of formulas were picked because they appear to meet the criteria of A-stability and stability at infinity, and were also of the same general form which facilitates including them in a variable order method:

$$0 = \sum_{i=0}^{k} \alpha_i^{(0)} y_{n-i} + h\, \alpha_0^{(1)} y'_n \qquad k = 1,2 \qquad (2.12)$$

$$0 = \sum_{i=0}^{k} \alpha_i^{(0)} y_{n-i} + \sum_{j=1}^{2} h^j \alpha_0^{(j)} y_n^{(j)} \qquad k = 2,3,4$$

$$0 = \sum_{i=0}^{k} \alpha_i^{(0)} y_{n-i} + \sum_{j=1}^{3} h^j \alpha_0^{(j)} y_n^{(j)} \qquad k = 4,5$$

$$0 = \sum_{i=0}^{k} \alpha_i^{(0)} y_{n-i} + \sum_{j=1}^{4} h^j \alpha_0^{(j)} y_n^{(j)} \qquad k = 5,6,7,8.$$

The formulas are of orders 1 through 11, the order being k+s-1, with $\varepsilon = s/k$ for ε-stability. The stability region generated by the graphical program for several of the above formulas appears in Figure 4. Since it might be possible that the boundary of the stability region crosses the imaginary axis and not be apparent on the plot, the roots $|\zeta_i(\mu)|$ for $\mu = iy$ with $y \in \{0.01n: n = 0(1)200\} \cup \{2+0.1n: n = 0(1)180\}$ were calculated and found to be less than 1. Thus, at least a numerical indication exists that the stability region doesn't cross the imaginary axis and that the formulas (2.12) are A-stable. Genin [24] has conjectured that the maximum order attainable for A-stable multivalue methods satisfies

$$p_{max} = \begin{cases} 2s+k-1 & s \geq k \\ 3s-1 & s \leq k. \end{cases} \qquad (2.13)$$

The cases (s=1, k=2), (s=2, k=4) and (s=4, k=8) attain this maximum order. A comment on his constructive demonstration for several values of s and k of this conjecture appears in section 2.3. It should be noted that when s > 1, these orders are higher than those attainable with the Obrechkoff's methods.

The coefficients $\alpha_i^{(j)}$ for each method are rational numbers computed by replacing $y_{n-i}^{(j)}$ by $y^{(j)}(t_{n-i})$ in each of formulas (2.12), expanding them in a power series about t_n, and equating coefficients of like powers of h for as high an order as possible. This leads to a linear system of k+s+1 equations for the unknowns $\alpha_i^{(j)}$. The matrix associated with this linear system, as described in section 3.1, is a variant of the Vandermonde

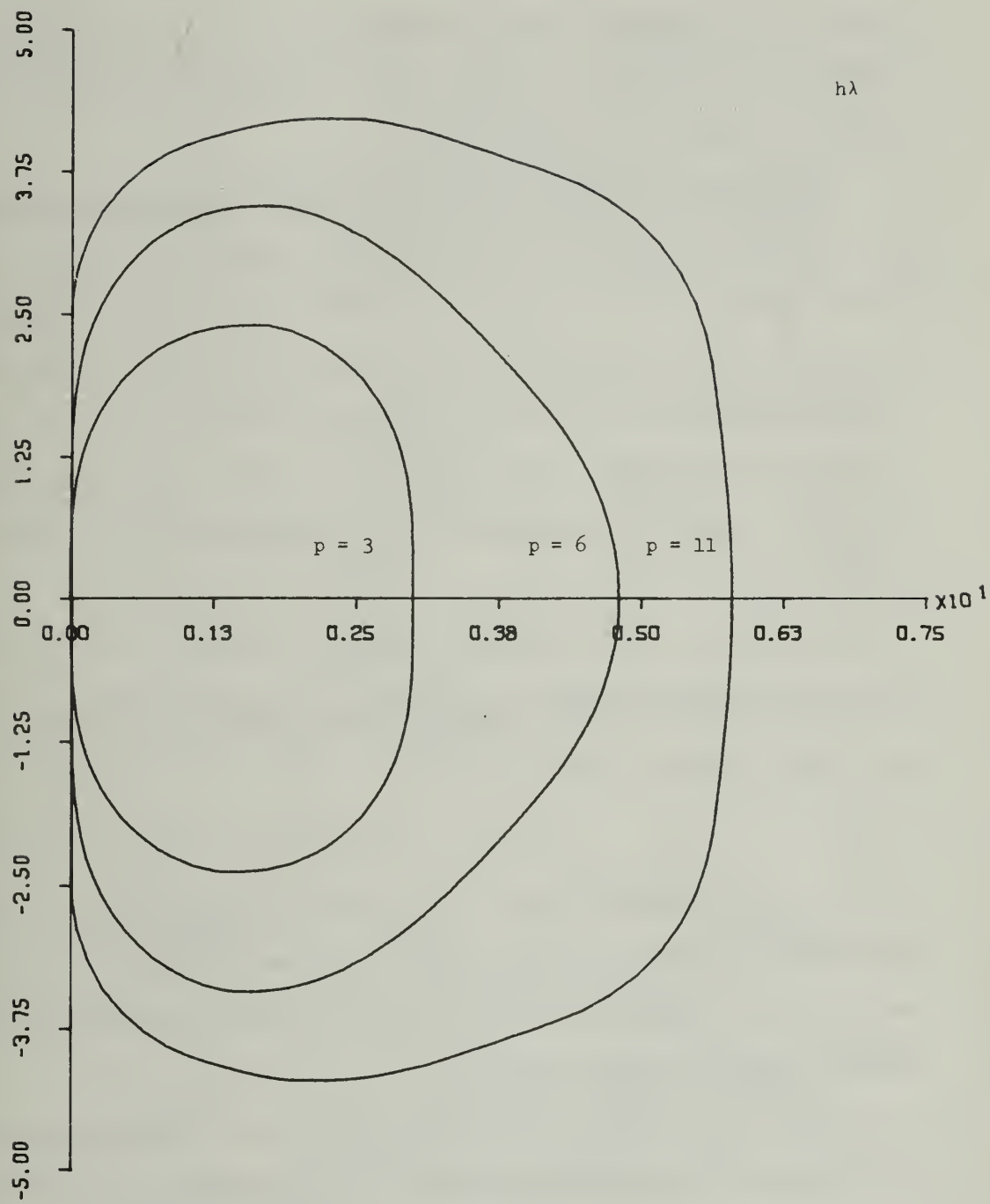

Figure 4. Stability regions for formulas (2.12) of order p for p equal to 3, 6, and 11. Formulas are stable on the exterior of the closed curve.

matrix and has a non-zero determinant, so the system has a solution. The search program described above computes the $\alpha_i^{(j)}$ as double precision floating point numbers. Computation of the coefficients as rational numbers is described in section 3.1.

2.3 Convergence and Error Estimation for Multi-derivative Formulas

Although the stability characteristics and consistency of the formulas (2.12) have been established, the standard result that stability, in the sense that $\rho(\xi)$ satisfies the strict root condition, and consistency are together sufficient conditions to guarantee convergence has not been established for methods of the form (2.1). This section will provide such a result. Further, an asymptotic error estimate will be provided for the global error

$$e_n = y_n - y(t_n) \tag{2.14}$$

for small stepsize h. The theorem statement yields the proper scaling for the local truncation error

$$d_n = y_n - y_{n-1}(t_n) \tag{2.15}$$

where $y_{n-1}(t)$ solves $y'_{n-1}(t) = f(y_{n-1}, t)$, $y_{n-1}(t_{n-1}) = y_{n-1}$. A note on the asymptotic error behavior of some formulas of Genin (1974) used to motivate his conjecture about the maximum order of A-stable multiderivative methods completes this chapter.

All of the formulas (2.12) are implicit, thus requiring the solution of a system of non-linear equations. If this system is solved iteratively then the method of the numerical solution will be called an Iterated Corrector method. Since any iterative procedure needs some initial estimate near the fixed point of the system, the way this estimate $\underline{y}_{n,(0)}$ to $\underline{y}_n$ is obtained is the method Predictor. The predictor can be as simple as

$$\underline{y}_{n,(0)} = \underline{y}_{n-1}$$

or can be the result of extrapolation from the previous computed values. Its exact form has no effect on the solution as long as $\underline{y}_{n,(0)}$ is in the radius of convergence of the iterative process and the corrector is iterated to convergence. If a bounded number of successive substitution corrections are used, the predictor affects the numerical solution through the truncation error.

Let an iterated corrector method be applied to the single equation

$$y' = f(y,t).$$

Let the values used by the method be stored in a vector

$$\underline{y}_n = (y_n, hy'_n, \ldots, h^s y_n^{(s)}, \ldots, y_{n-k}, \ldots, h^s y_{n-k}^{(s)})^T$$

where the vector elements are numbered from 0 and T is the transpose operator. Then a successive substitution iterated corrector method with an associated predictor may be formulated as:

Predictor: $$\underline{y}_{n,(0)} = B\underline{y}_{n-1}\ , \tag{2.16}$$

Corrector: $$F(\underline{y}_{n,(m)}) = \alpha_0^{(0)} y_{n,(m)} + \sum_{j=1}^{s} \alpha_0^{(j)} h^j f^{(j-1)}(y_{n,(m)})$$

$$+ \sum_{i=1}^{k} [\alpha_i^{(0)} y_{n-i} + \sum_{j=1}^{s} \alpha_i^{(j)} h^j f^{(j-1)}(y_{n-i})],$$

$$\underline{y}_{n,(m+1)} = \underline{y}_{n,(m)} + \underline{\ell}_0 F(\underline{y}_{n,(m)}) + \sum_{j=1}^{s} h^j \underline{\ell}_j \Delta f^{(j-1)}_{n,(m+1)},$$

$$\Delta f^{(i)}_{n,(m+1)} = \begin{cases} f^{(i)}(y_{n,(m)}) - f^{(i)}(y_{n,(m-1)}) & m > 0 \\ f^{(i)}(y_{n,(0)}) - (\underline{y}_{n,(0)})_{i+1} h^{-i-1} & m = 0 \end{cases}$$

Here $\underline{y}_{n,(m)}$ is the m-th corrector iterate of $\underline{y}_n$, and $y_{n,(m)} = (\underline{y}_{n,(m)})_0 = \underline{\ell}_0^T \underline{y}_{n,(m)}$ for $\underline{\ell}_0^T = (1,0,\ldots,0)$, coefficients are normalized so that

$\alpha_0^{(0)} = -1$, and the argument t has been omitted from f(y,t) when its value is clear. If the predictor B uses extrapolation, then its first s + 1 rows contain a generalized Hermite extrapolation operator for $y_{n+1}^{(j)}, j = 0,\ldots,s$ or the identity matrix otherwise, and the shift operator

$$b_{i,j} = \begin{cases} 1 & j = i - (s+1) \\ 0 & j \neq i - (s+1) \end{cases}$$

in the remaining rows. The i-th unit vector is $\underline{e}_i$. When iterated to convergence, this formulation is equivalent to any other formulation that solves the corrector exactly and will be used in proving theorem 2.1.

Theorem 2.1

Given

1) a consistent p-th order multiderivative formula (2.1) for which

$$\rho(\xi) = \sum_{i=0}^{k} \alpha_i^{(0)} \xi^{k-i} .$$

satisfies the strict root condition;

2) $y' = f(y,t)$ where f(y,t) is continuous for bounded y(t), $t_0 \leq t \leq t_f$, and each of $f_y^{(j-1)}(y,t_n)$, $j = 1,\ldots,s$, exists and is bounded by L_j, and $f^{(j-1)}(y,t_n)$ is continuous in y, for mesh points $t_n = t_0 + nh$, $n = 1,\ldots,N$;

3) a starting error $\underline{e}_0 = \underline{y}_0 - \underline{y}(t_0)$ bounded by

$$||\underline{e}_0|| \leq D'h^{p'} ;$$

then an iterated corrector method which is iterated to convergence (by any method) at each point t_n or which is iterated by (2.16) a bounded number of times (in which case, the truncation error of the predictor-

corrector formula must have p-th order) is convergent and there exist constants C_0, $\hat{C}_0$, C_2, D, h_0 dependent only on the method and f(y,t) such that

$$||\underline{e}_N|| \leq C_0 D' h^{p'} e^{C_2 b} + \hat{C}_0 D h^p e^{C_2 b}$$

for any $0 < h < h_0$, $b \geq Nh = t_f - t_0$, provided that the solution $y(t) \in C_{p+1}$, the space of functions with continuous (p+1)-th derivatives.

Proof

Express the local truncation error $\underline{d}_n$ as follows. Let $\underline{y}(t_n)$ be the correct value of $\underline{y}_n$, and let (2.16) be applied to $\underline{y}(t_{n-1})$ as

$$\tilde{\underline{y}}_{n,(0)} = B\, \underline{y}(t_{n-1}), \tag{2.17}$$

$$\tilde{\underline{y}}_{n,(m+1)} = \tilde{\underline{y}}_{n,(m)} + \underline{\ell}_0 F(\tilde{\underline{y}}_{n,(m)}) + \sum_{j=1}^{s} h^j \underline{\ell}_j \tilde{\Delta f}^{(j-1)}_{n,(m+1)},$$

$$\tilde{\underline{y}}_n = \tilde{\underline{y}}_{n,(M_n)}$$

where M_n is either bounded or is large enough that the corrector has converged. Then the local truncation error is

$$\underline{d}_n = \tilde{\underline{y}}_n - \underline{y}(t_n).$$

Here $\tilde{\Delta f}^{(i)}_{n,(m+1)}$ is defined for $f^{(i)}(\tilde{\underline{y}}_{n,(m)})$. Because the method has p-th order (if M_n is bounded, the predictor-corrector has p-th order), $\underline{d}_n$ can be expressed in a Taylor series with remainder terms in h^{p+1} times combinations of derivatives of f and y which can be bounded by the continuity hypothesis. Hence

$$||\underline{d}_n|| \leq D h^{p+1}.$$

Let

$$\underline{e}_n = \underline{y}_n - \underline{y}(t_n):$$

and

$$\underline{e}_{n,(m)} = \underline{y}_{n,(m)} - \tilde{\underline{y}}_{n,(m)}.$$

From the definitions

$$\underline{e}_{n,(0)} = \underline{y}_{n,(0)} - \tilde{\underline{y}}_{n,(0)} = B\,\underline{e}_{n-1},$$

$$\underline{e}_n = \underline{y}_n - \underline{y}(t_n) = \underline{y}_n - \tilde{\underline{y}}_n + \tilde{\underline{y}}_n - \underline{y}(t_n)$$

$$= \underline{e}_{n,(M_n)} + \underline{d}_n.$$

By subtracting (2.17) from (2.16) and using the mean value theorem, defining $f^{-1}(y) = y$ and omitting the argument of $\partial f/\partial y = f_y$ for notational convenience,

$$(\underline{e}_{n,(m+1)})_0 = [1 + \sum_{j=0}^{s} \alpha_0^{(j)} h^j f_y^{(j-1)}](\underline{e}_{n,(m)})_0 \qquad (2.18)$$

$$+ \sum_{i=1}^{k} \sum_{j=0}^{s} h^j \alpha_i^{(j)} f_y^{(j-1)} (\underline{e}_{n-i})_0,$$

and

$$h^{-j}(\underline{e}_{n,(m+1)})_j = y_{n,(m+1)}^{(j)} - \tilde{y}_{n,(m+1)}^{(j)} = f_y^{(j-1)}(\underline{e}_{n,(m)})_0. \qquad (2.19)$$

Note that the arguments of $f_y^{(j-1)}$ may not be the same in different appearances of $f_y^{(j-1)}$, but all we will need is that $||f_y^{(j-1)}||$ is bounded.

Therefore, since $\alpha_0^{(0)} = -1$ and $(\underline{e}_{n-i})_0 = \underline{\ell}_{(s+1)i}^T \underline{e}_{n,(m)}$

$$\underline{e}_n = S_n \underline{e}_{n-1} + \underline{d}_n \qquad (2.20)$$

where

$$S_n = \prod_{m=1}^{M_n} [\hat{I} + \sum_{j=0}^{s} h^j f_y^{(j-1)} \{\underline{\ell}_j \underline{\ell}_0^T \qquad (2.21)$$

$$+ \underline{\ell}_0 \sum_{i=0}^{K} \alpha_i^{(j)} \underline{\ell}_{(s+1)i}^T\}]B$$

$\hat{I}$ is the identity matrix with rows 0 through s zeroed, and $\underline{\ell}_j$ is the j-th unit vector.

First consider the case of bounded M_n. Since $|f^{(j-1)}| < L_j$, then

$$S_n = \hat{S}_0 + h \tilde{S}_n$$

where $\tilde{S}_n$ is a matrix whose elements are polynomials in h with bounded coefficients. The constant matrix

$$\hat{S}_0 = \begin{pmatrix} \hat{0} & \hat{\alpha}_1^{(0)} & \cdots & \hat{\alpha}_k^{(0)} \\ \hat{0} & & & \\ \vdots & & I & \\ \hat{0} & & & \end{pmatrix}^{M_n} \begin{pmatrix} \hat{B} & & \\ & & \hat{0} \\ & I & \vdots \\ & & \hat{0} \end{pmatrix}$$

where $\hat{\alpha}_i^{(0)}$ is the (s+1)x(s+1) array with $\alpha_i^{(0)}$ in the (0,0) position as the only nonzero element, $\hat{B}$ is the (s+1)x(k+1)(s+1) predictor operator, and $\hat{0}$ represents the (s+1)x(s+1) zero matrix. Due to the form of these operators, for any $M_n \geq 1$,

$$\hat{S}_0 = \begin{pmatrix} \hat{\alpha}_1^{(0)} & \cdots & \hat{\alpha}_k^{(0)} & \hat{0} \\ & & & \vdots \\ & I & & \\ & & & \hat{0} \end{pmatrix} \tag{2.22}$$

which is the companion matrix of a polynomial whose only non-zero roots are the roots of $\rho(\xi)$. Hence, 1 is the only eigenvalue of $\hat{S}_0$ with norm not less than 1, and there exists a constant C_0 such that for all $j \geq 0$

$$||\hat{S}_0^j|| \leq C_0.$$

If M_n, the number of iterations at each t_n, is not bounded allowing iteration to convergence, it can be shown that the coefficients of h in $\tilde{S}_n$ are still bounded for small enough h.

Let $\underline{\omega}_n$ be the total correction applied to $\underline{y}_{n,(0)}$ to get $\underline{y}_n$, that is

$$\underline{\omega}_n = \underline{y}_n - \underline{y}_{n,(0)}$$

Since the iteration has converged, we must have

$$F(\underline{y}_n) = 0$$

and

$$h^j f^{(j-1)}((\underline{y}_n)_0) = (\underline{y}_n)_j \qquad j = 1,\ldots,s \tag{2.23}$$

Similarly, let $\tilde{\underline{\omega}}_n$ be the change in $\tilde{\underline{y}}_{n,(0)}$ to get $\tilde{\underline{y}}_n$. $\tilde{\underline{y}}_n$ also satisfies equation (2.23). By the mean value theorem

$$\begin{aligned} 0 = F(\underline{y}_n) &= F(\tilde{\underline{y}}_n) + F_y(\underline{y}_n - \tilde{\underline{y}}_n) \\ &= F_y(\underline{y}_{n,(0)} - \tilde{\underline{y}}_{n,(0)} + \underline{\omega}_n - \tilde{\underline{\omega}}_n) \end{aligned}$$

and

$$h^j f_y^{(j-1)}(\underline{y}_n - \tilde{\underline{y}}_n)_0 = (\underline{y}_n - \tilde{\underline{y}}_n)_j \qquad j = 1,\ldots,s.$$

We can use equation (2.16) to write these as

$$0 = \sum_{i=0}^{k} \sum_{j=0}^{s} \alpha_i^{(j)} h^j f_y^{(j-1)} \underline{\ell}_{(s+1)i}^T (\underline{y}_{n,(0)} - \tilde{\underline{y}}_{n,(0)} + \underline{\omega}_n - \tilde{\underline{\omega}}_n) \tag{2.24}$$

and

$$\begin{aligned} 0 = \underline{\ell}_j^T &(\underline{y}_{n,(0)} - \tilde{\underline{y}}_{n,(0)} + \underline{\omega}_n - \tilde{\underline{\omega}}_n) \\ &- h^j f_y^{(j-1)} \underline{\ell}_0^T (\underline{y}_{n,(0)} - \tilde{\underline{y}}_{n,(0)} + \underline{\omega}_n - \tilde{\underline{\omega}}_n) \\ &j = 1,\ldots,s \end{aligned} \tag{2.25}$$

Note that all but the first s+1 components of $\underline{\omega}_n$ and $\tilde{\underline{\omega}}_n$ are zero so that $\underline{\ell}_j^T(\omega_n - \tilde{\omega}_n) = 0$ for $j > s$. Premultiply the equation (2.24) by $-\underline{\ell}_0$ and the equation (2.25) by $\underline{\ell}_j$, $j = 1,\ldots,s$; add, and add the equations $\{\sum_{j>s} \underline{\ell}_j \underline{\ell}_j^T + \underline{\ell}_0 \sum_{j=0}^{s} \sum_{i=1}^{k} h^j f_y^{(j-1)} \alpha_i^{(j)} \underline{\ell}_{(s+1)i}^T\}(\underline{\omega}_n - \tilde{\underline{\omega}}_n) = 0$ to get

$$(I - \sum_{j=1}^{s}(\underline{\ell}_j\underline{\ell}_0^T h^j f_y^{(j-1)} + \underline{\ell}_0\underline{\ell}_j^T \alpha_0^{(j)} h^j f_y^{(j-1)}))(\underline{\omega}_n - \tilde{\underline{\omega}}_n)$$

$$= (\sum_{i=0}^{k}\sum_{j=0}^{s} \underline{\ell}_0 \alpha_i^{(j)} h^j f_y^{(j-1)} \underline{\ell}_{(s+1)i}^T$$

$$- \sum_{j=1}^{s} \underline{\ell}_j\underline{\ell}_j^T + \sum_{j=1}^{s} \underline{\ell}_j\underline{\ell}_0^T h^j f_y^{(j-1)})(\underline{y}_{n,(0)} - \tilde{\underline{y}}_{n,(0)})$$

The matrix on the left-hand side can be inverted for all sufficiently small h because of the Lipschitz conditions on $f^{(j-1)}$, hence there exists an h_0 such that for $h \leq h_0$ the inverse of that matrix can be written as $I + hQ$ where Q is bounded. Since the right-hand side contains polynomials in h with bounded coefficients, we can write

$$\underline{\omega}_n - \tilde{\underline{\omega}}_n = (\sum_{i=0}^{k} \underline{\ell}_0\underline{\ell}_{(s+1)i}^T \alpha_i^{(0)} - \sum_{j=1}^{s} \underline{\ell}_j\underline{\ell}_j^T + h\hat{Q})(\underline{y}_{n,(0)} - \tilde{\underline{y}}_{n,(0)})$$

where $\hat{Q}$ has bounded coefficients. Hence

$$\underline{e}_n = \underline{y}_n - \tilde{\underline{y}}_n + \tilde{\underline{y}}_n - \underline{y}(t_n)$$

$$= \underline{y}_{n,(0)} - \tilde{\underline{y}}_{n,(0)} + \underline{\omega}_n - \tilde{\underline{\omega}}_n + \underline{d}_n$$

$$= [(I - \sum_{j=1}^{s} \underline{\ell}_j\underline{\ell}_j^T + \sum_{i=0}^{k} \underline{\ell}_0\underline{\ell}_{(s+1)i}^T \alpha_i^{(0)})B + h\hat{Q}B]\underline{e}_{n-1} + \underline{d}_n$$

$$= S_n\underline{e}_{n-1} + \underline{d}_n$$

where $S_n = \hat{S}_0 + h\tilde{S}_n$, $\tilde{S}_n$ is bounded for $h < h_0$, and $\hat{S}_0$ is identical to the previous $\hat{S}_0$ given in equation (2.22).

Then for $h < h_0$, there exist C_0, C_1, such that

$$||\hat{S}_0^j|| \leq C_0, \quad j \geq 1, \tag{2.26}$$

$$||\tilde{S}_n|| \leq C_1. \tag{2.27}$$

These satisfy the hypotheses of lemma 2.3 below, so

$$||S_m^n|| = ||S_{n-1}S_{n-2}\cdots S_m|| \leq C_0 e^{h(n-m)C_0C_1}. \tag{2.28}$$

For $t_N = t_f$,

$$\underline{e}_N = \sum_{j=1}^{N} S_{j+1}^{N} \underline{d}_j + S_1^{N+1} \underline{e}_0. \tag{2.29}$$

Thus,

$$\begin{aligned} ||\underline{e}_N|| &\leq N \, ||S_1^N|| \, ||\underline{d}_j|| + ||S_1^{N+1}|| \, ||\underline{e}_0|| \qquad (2.30) \\ &\leq \hat{C}_0 D e^{bC_2} h^p + C_0 D' e^{bC_2} h^{p'}. \end{aligned}$$

The constant terms are $b = Nh$, $N = (t_f - t_0)/h$, $\hat{C}_0 = C_0(t_f - t_0)$, $C_2 = C_0 C_1$. The error $||\underline{e}_N||$ goes to zero as $h \to 0$, $||\underline{e}_0|| \to 0$, and the formula is convergent. Q.E.D.

Lemma 2.3

If $S_n = \hat{S} + h\tilde{S}_n$ and there exist positive constants C_0, C_1, and h_0 such that for $h < h_0$,

1) $||\hat{S}^j|| \leq C_0, \quad j \geq 1,$

2) $||\tilde{S}_n|| \leq C_1.$

then

$$||S_m^n|| = ||S_{n-1} S_{n-2} \cdots S_m|| \leq C_0 \, e^{(n-m)hC_0C_1}. \tag{2.31}$$

Proof

It is true for $n = m$. Assume it is true for $m \leq j < n$. Then

$$||S_m^n|| \leq ||\hat{S}^{n-m}|| + h \sum_{j=m+1}^{n} ||\hat{S}^{n-j}|| \, ||\tilde{S}_{j-1}|| \, ||S_m^{j-1}||. \tag{2.32}$$

Note that

$$hC_0C_1 \leq (e^{hC_0C_1} - 1). \tag{2.33}$$

Replacing norms by bounds in (2.32) and using (2.33) yields

$$||S_m^n|| \leq C_0 + \sum_{j=m+1}^{n} C_0 (e^{hC_0C_1} - 1) e^{(j-m-1)hC_0C_1}. \tag{2.34}$$

Thus,

$$||S_m^n|| \leq C_0 e^{(n-m)hC_0C_1}, \tag{2.35}$$

and (2.31) is true by induction. Q.E.D.

There are several points to note about Theorem 2.1. First, the hypotheses are more strict than those for the convergence proof of 1-derivative multistep methods. In particular, the existence of $f_y^{(j-1)}$, $j = 1,\ldots,s$, and the continuity in y of $f^{(j-1)}(y,t_n)$ are required to assure the existence of S_n.

Second, the bounds shown for the global error are not useful in a computational sense because C_1 and thus C_2 are dependent on knowing the bounds $L^{(j)}$ on $f_y^{(j-1)}$ throughout the region of integration. Also, error bounds developed by replacing summations of N terms by N times the largest term, as in (2.30), tend to be very pessimistic.

An error estimate accurate for small h is more useful when trying to control the error in a computer calculation. Theorem 2.2 provides such an estimate. The following lemma will be used in proving Theorem 2.2.

Lemma 2.4

If

$$0 = \sum_{i=0}^{k} [\alpha_i^{(0)} y_{n-i} + \sum_{j=1}^{s} h^j \alpha_i^{(j)} f_{n-i}^{(j-1)}]$$

is a convergent method of order $r \geq 1$, and $\rho(\zeta)$ satisfies the strict root condition, then

$$0 = \sum_{i=0}^{k} [\alpha_i^{(0)} y_{n-i} + h\alpha_i^{(1)} f_{n-i}] \tag{2.36}$$

is a convergent method of order $r' \geq 1$.

Proof

Consider

$$L_h(y(t_n)) = \sum_{i=0}^{k}[\alpha_i^{(0)} y(t_{n-i}) + \sum_{j=1}^{s} h^j \alpha_i^{(j)} f^{(j-1)}(y(t_{n-i}),t_{n-i})] \tag{2.37}$$

for $y(t) = e^{\lambda t}$.

Then

$$L_h(e^{\lambda t}) = \sum_{i=0}^{k}[(\alpha_i^{(0)} + \sum_{j=1}^{s} h^j \lambda^j \alpha_i^{(j)}) e^{\lambda(t-ih)}]$$

$$= e^{\lambda(t-hk)}[\rho(e^{h\lambda}) + \sum_{j=1}^{s} (h\lambda)^j \sigma_j(e^{h\lambda})]. \tag{2.38}$$

Since the method is of order r

$$L_h(e^{\lambda t}) = C_{r+1}(h\lambda)^{r+1} e^{\lambda(t-hk)} + O(h^{r+2}),$$

Hence

$$\rho(e^{h\lambda}) + \sum_{j=1}^{s} (h\lambda)^j \sigma_j(e^{h\lambda}) = C_{r+1}(h\lambda)^{r+1} + O(h^{r+2}). \tag{2.39}$$

Writing $h\lambda = \log(1+z)$ and using $h\lambda = z + O(z^2)$, we see that

$$\rho(1+z) + \sum_{j=1}^{s} (\log(1+z))^j \sigma_j(1+z) = C_{r+1} z^{r+1} + O(z^{r+2}) \tag{2.40}$$

is a necessary and sufficient condition for the method to be order r.

Expand this in a power series about 1 in z;

$$\rho(1) + z\rho'(1) + z\sigma_1(1) + O(z^2) = C_{r+1} z^{r+1} + O(z^{r+2}) \tag{2.41}$$

so order $r \geq 1 \Rightarrow \rho(1) = 0$ and $\rho'(1) + \sigma_1(1) = 0$. Thus, a method involving only $\rho(\xi)$ and $\sigma_1(\xi)$ will be consistent. If $\rho(\xi)$ satisfies the strict root condition, such a method will also be strongly stable and thus convergent. So

$$\sum_{i=0}^{k} (\alpha_i^{(0)} y_{n-i} + h\alpha_i^{(1)} f_{n-i}) \tag{2.42}$$

defines a convergent method of order $r' \geq 1$. Q.E.D.

We can now state the theorem describing the behavior of the global error for a small but non-zero stepsize. The result will also yield the

proper scaling for the discretization error $L_h(y(t_n))$ to give the local truncation error (2.15), where

$$d_n = y_n - y_{n-1}(t_n).$$

Theorem 2.2

If a strongly stable convergent pth order multivalue formula of form (2.1) is applied to

$$y' = f(y,t),$$

$$y(t_0) = y_0,$$

and $f^{(p)}(y,t)$ is continuously differentiable with respect to t, and $f_y^{(j)}(y,t)$, $j = 0,\ldots,s-1$, is continuous in y for bounded y in the range of t, and all starting values are exact, then

$$e_n = h^p \delta(t_n) + O(h^{p+1}),$$

where $\delta(t)$ satisfies

$$\delta' = f_y(y,t)\ \delta(t) + \frac{C_{p+1}}{\sum_{i=0}^{k} \alpha_i^{(1)}}\ y^{(p+1)}(t),$$

$$\delta(t_0) = 0.$$

Proof

Consider the discretization error which, from the assumption on f(p), is

$$L_h(y(t_n)) = \sum_{i=0}^{k}[\alpha_i^{(0)} y(t_{n-i}) + \sum_{j=1}^{s} \alpha_i^{(j)}\ f^{(j-1)}(y(t_{n-i}),t_{n-i})] \quad (2.43)$$

$$= C_{p+1}\ h^{p+1} y^{(p+1)}(t_n) + O(h^{p+2}).$$

The corrector iterated to convergence satisfies

$$0 = \sum_{i=0}^{k}[\alpha_i^{(0)} y_{n-i} + \sum_{j=1}^{s} h^j\ \alpha_i^{(j)} f_{n-i}^{(j-1)}]. \quad (2.44)$$

Subtracting (2.43) from (2.44) gives

$$\sum_{i=0}^{k}[\alpha_i^{(0)} e_{n-i} + \sum_{j=1}^{s} h^j \alpha_i^{(j)}(f_{n-i}^{(j-1)} - f^{(j-1)}(y(t_{n-i}),t_{n-i})] + \quad (2.45)$$

$$C_{p+1}h^{p+1}y^{(p+1)}(t_n) = O(h^{p+2}).$$

Assume the theorem is true for e_0 through e_{n-1}. The mean value theorem gives

$$f_{n-i}^{(j-1)} - f^{(j-1)}(y(t_{n-i}),t_{n-i}) = f_y^{(j-1)}(y(t_{n-i})+\theta,t_{n-i})e_{n-i} \quad (2.46)$$

for some θ satisfying $0 \leq |\theta| \leq |e_{n-i}|$. Since by theorem 2.1

$$e_{n-i} = O(h^p),$$

(2.46) can be written as

$$f_{n-i}^{(j-1)} - f^{(j-1)}(y(t_{n-i}),t_{n-i}) = f_y^{(j-1)}(y(t_{n-i}),t_{n-i})e_{n-i} + O(h^{2p}). \quad (2.47)$$

Thus,

$$\sum_{i=0}^{k}[\alpha_i^{(0)} e_{n-i} + h\alpha_i^{(1)} f_y(y(t_{n-i}),t_{n-i})e_{n-i}] + C_{p+1}h^{p+1}y^{(p+1)}(t_n)$$

$$= \sum_{i=0}^{k}\sum_{j=2}^{s} h^j \alpha_i^{(j)} f_y^{(j-1)}(y(t_{n-i}),t_{n-i}) + O(h^{p+2}). \quad (2.48)$$

If

$$e_n = h^p\delta(t_n) + O(h^{p+1}),$$

then

$$\sum_{i=0}^{k}[\alpha_i^{(0)}\delta_{n-i} + h\alpha_i^{(1)}(f_y(y(t_{n-i}),t_{n-i})\delta_{n-i} + \frac{C_{p+1}}{\sum_{i=0}^{k}\alpha_i^{(1)}} y^{(p+1)}(t_{n-i}))]$$

$$= O(h^2). \quad (2.49)$$

From lemma 2.4, this is a convergent method which solves

$$\delta'(t) = f_y(y(t),t)\delta(t) + \frac{C_{p+1}}{\sum_{i=0}^{k}\alpha_i^{(1)}} y^{(p+1)}(t). \quad (2.50)$$

Q.E.D.

Under the more realistic assumptions that the starting values are not exact, a similar result can be shown, following Henrici [26: sec 5.3]. Because of (2.50) the appropriate quantity to estimate and control is

$$d_n = h^{p+1} \frac{C_{p+1}}{\sum_{i=0}^{k} \alpha_i^{(1)}} y^{(p+1)}(t). \tag{2.51}$$

This will be discussed further in the next chapter in relation to an actual computer program.

Genin [24] has conjectured that the maximum order attainable by an A-stable k-step s-derivative method is

$$p_{max} = \begin{cases} 2s + k - 1 & k \leq s \\ 3s - 1 & k \geq s \end{cases} . \tag{2.52}$$

To motivate this conjecture, he develops canonical multiderivative formulas for the linear multistep (s = 1) and linear multiderivative (k = 1) formulas and demonstrates that p_{max} is attainable as in (2.52) for k-step s-derivative formulas that are products of these canonical formulas. The analysis depends on an algebraic manipulation of the multiderivative multistep formulas to yield a canonical polynomial H(p,q) in the variables $p = (\xi+1)/(\xi-1)$ and $q = h\lambda$ corresponding to the characteristic polynomial

$$\rho(\xi) + \sum_{i=1}^{s} (h\lambda)^i \sigma_i(\xi) = 0.$$

He proves that a formula (2.1) is A-stable if and only if the canonical polynomial

$$H(p,q) = \sum_{i=0}^{s} \sum_{j=0}^{k} a_{ij} p^{k-j} q^i = 0 \tag{2.53}$$

is a two variable Hurwitz polynomial in the narrow sense. He also provides an order characterization based on H(p,q). If $H_s(p,q)$ is the canonical polynomial of the unique optimal highest order A-stable linear multi-

derivative formula, then for all $k < s$, the canonical polynomial

$$H_{k,s}(p,q) = H_{s-(k-1)}(p,q)[H_1(p,q)]^{k-1} \qquad k < s$$

is an A-stable k-step s-derivative formula of order 2s+k-1. If $G_k(p,q)$ is an A-stable 1-derivative k-step formula of order 2, the maximum attainable order for multistep formulas, then $G_1(p,q) = H_1(p,q)$ is unique. For all $k \geq s$

$$H_{k,s} = G_{k-(s-1)}(p,q)[H_1(p,q)]^{s-1} \qquad k \geq s$$

is an A-stable k-step s-derivative formula of order 3s-1. In both these cases the order is given on the basis of the first nonzero term of the expansion of the truncation error. However,

$$H_1(p,q) = 2 + pq$$

corresponds to the trapezoidal formula and $\rho(\xi) = (\xi-1)$. Therefore, for $k < s$, $H_{k,s}(p,q)$ has a characteristic subpolynomial

$$\rho(\xi) = (\xi-1)^k\hat{\rho}(\xi) = (\xi-1)^{r+1}\hat{\rho}(\xi),$$

and for $k \geq s$,

$$\rho(\xi) = (\xi-1)^s\tilde{\rho}(\xi) = (\xi-1)^{r+1}\tilde{\rho}(\xi),$$

for some $\hat{\rho}$, $\tilde{\rho}$, and r dependent on k and s. All the roots of $\hat{\rho}(\xi)$ and $\tilde{\rho}(\xi)$ are less than 1 in norm.

Therefore, the formulas of order p_{max} are not stable in the sense that $\rho(\xi)$ satisfies the root condition which requires the roots on the unit circle to be simple. If they were used in a calculation, the computed solution would be very sensitive to both roundoff and truncation errors. Also, note that the numerical solution to

$$y' = \lambda e^{\lambda t}, \; y(0) = 1,$$

which meets all the hypotheses of theorem 2.1 except for the multiple

root of 1, has a global error that is different from that shown in the theorem by a factor of h^{-r} where

$$r = \begin{cases} k-1 & k < s \\ s-1 & k \geq s \end{cases}$$

This can be seen by observing that the method truncation error is

$$d_n = \sum_{i=1}^{\infty} C_{v+r+i}(h\lambda)^{v+r+i} e^{\lambda t_n} .$$

Then the multiderivative multistep formula has the solution

$$y_n = \frac{\sum_{j=1}^{s} (h\lambda)^j \sigma_j(e^{h\lambda}) e^{\lambda t_n}}{-\rho(e^{h\lambda})} + \frac{K_n (h\lambda)^{v+r+1}}{-\rho(e^{h\lambda})} \qquad (2.54)$$

where

$$K_n = \sum_{i=1}^{\infty} (h\lambda)^{i-1} C_{v+r+i} \sum_{m=0}^{k-1} \alpha_m^{(0)} \sum_{j=0}^{k-i-1} e^{jh\lambda} e^{\lambda t_{n+k-j-i}}.$$

Since the difference operator for the solution $e^{\lambda t}$ satisfies

$$L_h(e^{\lambda t_n}) = [\rho(e^{h\lambda}) + \sum_{j=1}^{s} (h\lambda)^j \sigma_j(e^{h\lambda})] e^{\lambda t_{n-k}}$$

$$= C_{v+r+1}(h\lambda)^{v+r+1} e^{\lambda t_{n-k}} + O(h^{v+r+2}),$$

then

$$\sum_{j=1}^{s} [(h\lambda)^j \sigma_j(e^{h\lambda})]/-\rho(e^{h\lambda}) = 1 - \frac{C_{v+r+1}(h\lambda)^{v+r+1}}{\rho(e^{h\lambda})} \qquad (2.55)$$

$$+O(h^{v+r+2}/\rho(e^{h\lambda}))$$

Expanding $e^{h\lambda}$ in a Maclaurin series and using it in $\rho(e^{h\lambda})$ gives

$$\rho(e^{h\lambda}) = \alpha_0^{(0)}(h\lambda)^{r+1} \Sigma$$

where Σ is an infinite sum in $h\lambda$ with a constant first term. Hence from (2.54)

$$y_n - e^{\lambda t_n} = \frac{-C_{v+r+1}(h\lambda)^v e^{t_n} - K_n(h\lambda)^v}{\alpha_0^{(0)}\ \Sigma} + O(h^{v+1})$$

which is the expected form of the global error of a method of order r lower than what is expected from the truncation error. Thus, theorem 2.1 does not apply for the above formulas which attain p_{max}; they appear to achieve that order only in the indicated truncation error, are not stable under the usual definition, and are very sensitive to error. In fact, the apparent error for all these formulas when integrating $y' = e^{\lambda t}$ is

$$p_{app} = \begin{cases} 2s + k - 1 - (k-1) & k < s \\ 3s - 1 - (s-1) & k \geq s \end{cases}$$

$$= 2s \quad ,$$

the same as for Obrechkoff's methods. Genin [24] shows that for (k=2,s=2) (k=3,s=2), and (k=2,s=3) there are no methods with order greater than p_{max} by applying the Hurwitz criterion to $H_{k,s}(p,q)$. However, all three of the formulas given that achieve the expected maximum order have

$$\rho(\xi) = (\xi - 1)^2\hat{\rho}(\xi)$$

and therefore have a global error of a formula of order 2s for the equation $y' = \lambda e^{\lambda t}$. It should be noted, however, that p_{max}, as in (2.52), is attained by formulas (2.12) for (k=2,s=1), (k=4,s=2), and (k=8,s=4). This suggests that the conjecture is possibly correct; only the supporting formulas in [24] are inapplicable.

3. Implementation

A computer program for a variable step, variable order method for the numerical solution of ordinary differential equations is based on four elements:

1) A set of formulas - to compute the next approximation, and descriptions of how such formulas are to be applied;
2) An estimator - of the local error committed at each step;
3) An acceptance rule - to determine whether to accept a value from a single step or try again with a smaller stepsize;
4) A stepsize strategy - to determine the best next stepsize as well as how often to change it.

In a variable order method, the best stepsize is picked from that used by several different orders, so the next order determination is part of the stepsize strategy.

Hull [28] and Ehle [15] consider these attributes of different methods in relation to specific problems in order to prove effectiveness and compare numerical behavior, respectively.

This chapter will define a method for solution of stiff ordinary differential equations by providing these four elements. To avoid notational complexity, all references will be to the equation

$$y' = f(y,t)$$

rather than to a system of equations. Where evaluation of a system will require greater complexity, such as for the Jacobian f_y, which is a matrix for systems, such complexity will be noted.

3.1. The Iterated Corrector Method

All of the formulas (2.12) are maximally implicit which makes them

suitable as corrector formulas in an iterated corrector method, as defined in section 2.2. By adding an extrapolating predictor formula of the same order and choosing an efficient vector representation $\underline{a}_n$ for the saved values of the dependent variable, an iterated corrector formulation is provided as the method calculator. A form for representing the dependent variables proposed by Nordsieck [35] provides a convenient extrapolative predictor as well as an efficient means for manipulation of the stored values.

The set of saved values $hy_n^{(1)},\ldots, h^s y_n^{(s)}, y_{n-i}$, $i = 1,\ldots,k$, used to compute y_n in the k-step s-derivative formulas (2.12) uniquely determines a k+s-1 degree polynomial which interpolates these values. This polynomial can be equally well represented by its value y_n and its first k+s-1 derivatives at t_n. The Nordsieck method stores the elements of the Taylor series of this polynomial about t_n. If

$$\underline{y}_n = (y_n, y_{n-1}, \cdots, y_{n-k+1}, hy_n^{(1)}, \ldots, h^s y_n^{(s)})^T$$

and

$$\underline{a}_n = (y_n, hy_n^{(1)}, h^2 y_n^{(2)}/2, \ldots, h^q y_n^{(q)}/q!)^T,$$

where $q = k+s-1$, T is the transpose operator, and y_n is the 0-th element of both $\underline{a}_n$ and $\underline{y}_n$, then

$$\underline{a}_n = Q\,\underline{y}_n \tag{3.1}$$

where Q is a non-singular matrix. Further,

$$\underline{a}_{n+1} = P\underline{a}_n + O(h^{q+1}) \tag{3.2}$$

where P is the Pascal triangle matrix

$$\{P_{ij}\} = \{\binom{j}{i}\}.$$

Thus, (3.2) can be used as a q-th order predictor since

$$\frac{h^i y_n^{(i)}}{i!} = \sum_{j=i}^{q} \binom{j}{i} \frac{h^j y_{n-1}^{(j)}}{j!} + O(h^{q+1}).$$

The efficiency of storing the dependent variables in such a form arises from two considerations. First, the predictor step can be calculated using only additions of elements of $\underline{a}_n$ in the same way as the Pascal triangle matrix can be formed. Second, when a new stepsize has been chosen, the new variable can be formed by

$$\tilde{\underline{a}}_n = \begin{pmatrix} 1 & & & & \\ & \alpha & & & \\ & & \alpha^2 & & \\ & & & \ddots & \\ & & & & \alpha^q \end{pmatrix} \underline{a}_n$$

for $\alpha = \tilde{h}/h$. This essentially performs a new q-th order expansion about t_n which is equivalent to an approximation to a new set of data $y_n, \tilde{h}y_n^{(1)}, \ldots, \tilde{h}^s y_n^{(s)}, \tilde{y}_{n-1}, \ldots, \tilde{y}_{n-k}$. Although $y(t_n - j\tilde{h})$ may never have been approximated by some $\tilde{y}_{n-j}$, it is still within $O(h^{q+1})$ of the correct value by the properties of the Taylor series. As will be discussed in section 3.4, this does not affect the stability of the method if stepsize and order changes are restricted by certain criteria.

Define the change in y_n necessary at each corrector iteration by

$$F(\underline{y}_n,(m)) = -y_{n,(m)} + \Sigma + \sum_{j=1}^{s} h^j \alpha_0^{(j)} f^{(j-1)}(y_{n,(m)}, t_n),$$

$$\Sigma = \sum_{i=1}^{k} \alpha_i^{(0)} y_{n-i},$$

$$y_{n,(m)} = (\underline{y}_{n,(m)})_0. \tag{3.3}$$

The subscript in $(\underline{y}_n)_i$ denotes the i-th element of $\underline{y}_n$ (the initial

element being $(\underline{y}_n)_0 = y_n$). The iterated corrector formula is then

$$\underline{y}_{n,(0)} = B\,\underline{y}_{n-1},$$

$$\underline{y}_{n,(m+1)} = \underline{y}_{n,(m)} + \underline{e}_0 F(\underline{y}_{n,(m)}) + \sum_{j=1}^{s} h^j \underline{e}_j \Delta f^{(j-1)}_{n,(m+1)}, \tag{3.4}$$

$$\Delta f^{(i)}_{n,(m+1)} = \begin{cases} f^{(i)}(y_{n,(m)}) - f^{(i)}(y_{n,(m-1)}) & m > 0 \\ f^{(i)}(y_{n,(0)}) - (\underline{y}_n)_i & m = 0 \end{cases}.$$

B is the matrix $Q^{-1}PQ$. The subscript n denotes $\underline{y}_n \doteq \underline{y}(t_n)$, and (m) denotes the m-th corrector iterate. The vector $\underline{e}_i$ is the i-th unit vector, where

$$\underline{e}_0 = (1,0,\ldots,0)^T.$$

This can be converted to Nordsieck's form as

$$\underline{a}_{n,(0)} = P\,\underline{a}_{n-1} = QBQ^{-1}\,\underline{a}_{n-1},$$

$$\underline{a}_{n,(m+1)} = \underline{a}_{n,(m)} + \underline{\ell}_0 G(\underline{a}_{n,(m)}) + \sum_{j=1}^{s} \ell_j \Delta f^{(j-1)}_{n,(m+1)},$$

$$G(\underline{a}_{n,(m)}) = F(Q^{-1}\,\underline{a}_{n,(m)}),$$

$$\underline{\ell}_0 = \underline{e}_0,$$

$$\underline{\ell}_j = Q(\alpha_0^{(j)}\,\underline{e}_0 + \underline{e}_j), \qquad j = 1,\ldots,s. \tag{3.5}$$

The derivation of the coefficients $\alpha_i^{(j)}$ and $\underline{\ell}_j$ can now be described more fully. To get $\alpha_i^{(j)}$ for a formula (2.12) requires the solution of the rational system

$$\begin{pmatrix} -1 & -2 & \cdots & -k & 0 \\ \frac{(-1)^2}{2!} & \frac{(-2)^2}{2!} & \cdots & \frac{(-k)^2}{2!} & I \\ & & & & \\ \frac{(-1)^q}{q!} & \frac{(-2)^q}{q!} & & \frac{(-k)^q}{q!} & 0 \end{pmatrix} \begin{pmatrix} \alpha_1^{(0)} \\ \vdots \\ \alpha_k^{(0)} \\ \alpha_0^{(1)} \\ \\ \alpha_0^{(s)} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \\ 0 \end{pmatrix} \tag{3.6}$$

after scaling the $\alpha_i^{(j)}$ so that $\alpha_0^{(0)} = -1$. I is the s x s identity matrix. For finding the $\underline{\ell}_i$ coefficients, note that

$$Q^{-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & -1 & \cdots & (-1)^q \\ \vdots & & & \\ 1 & (-k+1) & \cdots & (-k+1)^q \\ \\ 0 & 1! & 0 & \cdots & 0 \\ 0 & 0 & 2! & \cdots & 0 \\ 0 & & & \cdots & 0 \end{pmatrix}$$

is also rational. The vectors $\underline{\ell}_i$, i = 1,...,s, can be found by solving

$$Q^{-1}\underline{\ell}_i = (\alpha_0^{(i)}\underline{e}_0 + \underline{e}_1).$$

A program was written in ALTRAN, a formula manipulation language from Bell Laboratories described in [7], to set up these rational matrices and vectors and solve for the coefficients using an algebraic linear equation solver ASOLVE which is a resident language routine. The results for $(i!)\alpha_0^{(i)}$, i = 1,...,s, and $\underline{\ell}_i$ appear in the comments accompanying the program D4 in Appendix I for formulas (2.12) up through order 8. The rational results for larger orders could not be computed by the ALTRAN program due to overflow in the rational representation of data.

Another point of interest is the method for calculating Σ, a linear combination of previous computed values y_{n-i}. If this computation were done with the exact values stored in $\underline{y}_n$, then at every change in stepsize either other calculated values of y_{n-i} must be available, thus restricting the program to integral multiples on increasing stepsize and starting over at the lowest order on decreasing stepsize as in [16], or else interpolating new points. The stored values of the dependent variables form the Taylor expansion of an interpolating polynomial $P(t)$ which, for all t in $t_0 \leq t \leq t_f$, satisfies

$$0 = \sum_{i=0}^{k} \alpha_i^{(0)} P(t-ih) + \sum_{j=1}^{s} \alpha_0^{(j)} h^j P^{(j)}(t),$$

$$P(t_n - ih) = y_{n-i}, \qquad i = 1,\ldots,k+1,$$

$$P^{(j)}(t_{n-1}) = f^{(j-1)}(y_{n-1}, t_{n-1}), \qquad j = 1,\ldots,s.$$

Hence the formula

$$\Sigma = \sum_{i=1}^{k} \alpha_i^{(0)} y_{n-i} = -\sum_{j=1}^{s} \alpha_0^{(j)} h^j y_{n,(0)}^{(j)} + y_{n,(0)} \tag{3.7}$$

is exact. Formula (3.7) is used in calculating Σ immediately after the predictor step in D4.

Although the restriction on $h\lambda$ due to the absolute stability requirement on the formula has been removed for $Re(h\lambda) < 0$, the corrector may still not converge for the large values of h which the program may attempt to use. This can be overcome by considering a Newton's iteration (or quasi-Newton) on the 0-th term of the vector $\underline{a}_n$. The function whose zero is sought is

$$F(\underline{y}) = (-\underline{y})_0 + \Sigma + \sum_{j=1}^{s} h^j \alpha_0^{(j)} f^{(j-1)}((\underline{y})_0, t)$$

where $(\underline{y})_0$ is the unknown y_n, Σ and t remain constant, and the $f^{(j-1)}$ are considered as function of y_n. The corrector iteration on $y_{n,(m)} = (\underline{y}_{n,(m)})_0$ then becomes

$$y_{n,(m+1)} = y_{n,(m)} - [\frac{dF}{dy}]^{-1} F(\underline{y}_{n,(m)}) \tag{3.8}$$

$$= y_{n,(m)} + [I - \sum_{j=1}^{s} h^j \alpha_0^{(j)} f_y^{(j-1)}(y_{n,(m)}, t_n)]^{-1} F(\underline{y}_{n,(m)}).$$

This leads to the iterated corrector formula with extrapolative predictor

$$\underline{a}_{n,0} = P\, \underline{a}_{n-1} \tag{3.9}$$

$$\underline{a}_{n,(m+1)} = \underline{a}_{n,(m)} + \underline{\ell}_0 W F(\underline{a}_{n,(m)}) + \sum_{j=1}^{s} \underline{\ell}_j \Delta f^{(j-1)}_{n,(m+1)}$$

$$W = [I - \sum_{j=1}^{s} h^j \alpha_0^{(j)} f_y^{(j-1)}(y_{n,(m)}, t_n)]^{-1}.$$

Because of the expense of evaluating W, which is a matrix for solving systems of equations, it is evaluated as seldom as possible, since when the resulting quasi-Newton method (3.9) converges, it converges to a solution of (3.5). In fact, W can be computed by numerical differencing with respect to $\underline{y}$ and still give the desired convergence. Since the proof of convergence in Theorem 2.1 assumed only that the corrector was solved, the fact that Newton's method is used does not affect the proof.

3.2 Stepsize and Order Changing

Let $\hat{C}_{q+1} = \dfrac{C_{q+1}}{\alpha_0^{(1)}}$

where C_{q+1} and $\alpha_0^{(1)}$ are associated with the formula of order q in (2.12). Then the truncation error per step in the asymptotic error estimate of Theorem 2.2,

$$d_n = \hat{C}_{q+1} h^{q+1} y_n^{(q+1)} + O(h^{q+2}), \qquad (3.11)$$

is used in the stepsize and order changing algorithm. An estimate E_n of d_n will be made and this will be compared to the desired error EPS set by the user to determine whether or not the step will be accepted. This can also be used to determine what stepsize $\hat{h}_q$ should have been used instead of h_q to get d_n = EPS for the qth order method used whenever it is possible or necessary to change the stepsize. In practice, $\hat{h}_q$ will be chosen smaller than the optimum value computed to account for the error in E_n.

To derive this algorithm, note that the predictor step of (3.9) does not modify

$$(\underline{a}_n)_q = \frac{h^q y_n^{(q)}}{q!},$$

the last element of the (q+1) length vector $\underline{a}_n$, and thus its final corrector component

$$\sum_{m=1}^{M_n} \sum_{j=1}^{s} (\underline{\ell}_j)_q \Delta f_{n,(m)}^{(j-1)}$$

is a backward difference

$$\frac{h^q y_n^{(q)}}{q!} - \frac{h^q y_{n-1}^{(q)}}{q!} = \frac{h^{q+1}}{q!} y_n^{(q+1)} + O(h^{q+2}). \qquad (3.12)$$

If, after the corrector converges and the error estimator satisfies

$$E_n = \hat{C}_{q+1}\, q! \,\Big\| \sum_{j=1}^{s} (\underline{\ell}_j)_q \left(\frac{h^j}{j!} f_n^{(j-1)} - e_j^T P\, \underline{a}_{n-1}\right) \Big\| < \varepsilon \qquad (3.13)$$

for ε = EPS, then the desired stepsize for a method at order q approximately satisfies

$$\hat{h}_q^{q+1} \phi_n = \varepsilon,$$

and the current stepsize h nearly satisfies

$$h_q^{q+1} \phi_n = E_n,$$

where ϕ_n is the principal error function $\hat{C}_{q+1} y_n^{(q+1)}$.

Therefore,

$$\hat{h}_q = h_q \left(\frac{\varepsilon}{E_n}\right)^{1/q+1}. \tag{3.14}$$

Thus, the asymptotic error estimate is used to decide if a value of $\underline{a}_n$ will be accepted, and to compute the best stepsize at order q. For possible order changing, the best stepsize is computed at convenient orders, in this case, q+1 and q-1, and the order corresponding to the largest stepsize is used. The computations for one order lower is

$$E_n^- = \hat{C}_q \, q! \, (\underline{a}_n)_q, \tag{3.15}$$

$$\hat{h}_{q-1} = h_q \left(\frac{\varepsilon}{E_n^-}\right)^{1/q}.$$

An estimate for $h^{q+2} y_n^{(q+2)}$ can be obtained by differencing E_n to get E_n^+:

$$E_n^+ = \frac{\hat{C}_{q+2}}{\hat{C}_{q+1}} (E_n - E_{n-1}) = \hat{C}_{q+2} \, h^{q+2} \, y_n^{(q+2)} + O(h^{q+3}), \tag{3.16}$$

$$\hat{h}_{q+1} = h_q \left(\frac{\varepsilon}{E_n^+}\right)^{1/q+2}.$$

By dividing the calculated values $\hat{h}_q$ by 1.2 and $\hat{h}_{q-1}$, $\hat{h}_{q+1}$, by 1.3, the computed next value takes into account the inaccuracy of the approximation to the asymptotic error estimate and also projects a bias toward leaving the order at q, since changing order has some overhead associated with it. Also, no stepsize increase of less than $.1h_q$ is allowed because of the overhead involved. When the estimated error is too large, only the best stepsizes at orders q and q-1 are considered.

The four elements of a variable stepsize, variable order multivalue method have now been determined. The calculator is (3.9). The estimator is (3.13). The acceptance rule (3.13) depends on the estimate of d_n by E_n. The stepsize strategy combines the stepsize and order calculations (3.14), (3.15), and (3.16). The exact algorithm involved when changing order will be considered in section 3.4. The variable representation is the Nordsieck form $\underline{a}_n$ which interacts with the above error estimators to provide access to the higher derivatives needed, to predict $\underline{a}_{n,(0)}$, and to provide an easy means of changing $\underline{a}_n$ with changing stepsize.

3.3 Program Description

A program D4 has been written to incorporate formulas (2.12) and the previously presented representations and algorithms into a FORTRAN subroutine package. The organization of the program and particularly the form of the stepsize and order changing procedures are very similar to the program DIFSUB of Gear [21]. The heuristic devices employed therein are used extensively throughout D4, which should result in approximately the same proportion of program overhead in the various types of calls to D4, i.e., successful steps, steps that don't converge on the corrector iteration, and steps that converge but don't meet the error criterion. Some major features of the program are outlined below and the program is listed with comments on its use in Appendix I.

The user is responsible for providing the differential equations in the form of a subroutine

```
DIFFUN(T,H,Y,DYS,NORD)
```

which places the first NORD elements, excluding the 0-th, of the Taylor series for Y about T into DYS. A routine is provided to calculate W by

numerical differencing, but the user may provide a routine to calculate W exactly if desired. In the linear case when $\underline{f}_y^{(i)} = A^{i+1}$ for $\underline{y}' = A\underline{y}$ the exact calculation will take less execution time than the N calls to DIFFUN for an N-dimensional system needed for numerical differencing; in the non-linear case, exact $\underline{f}_y^{(i)}$ may improve the rate of convergence.

Each call to D4 takes one step in the independent variable T and returns. The input value of stepsize H is tried first and reduced if necessary; the output value is the best stepsize for the next step. This may be changed by the calling program but an increase in H will probably not be accepted.

The program keeps internal variables from one step to the next and this prevents the use of D4 to solve more than one problem using alternating calls.

3.4 Properties of Stepsize and Order Changing Methods

All of the theory on stability and convergence presented in Chapter 2 and sections 3.1 and 3.2 were developed on the assumption that the order remains constant throughout the calculation and the stepsize is such that the values y_{n-i} are available as calculated. It is possible that the interpolatory change in $\underline{a}_n$ when the stepsize is changed might introduce factors which will prevent the method from being stable and convergent. No algorithm was presented for changing the order, and factors involved in changing the order which also might disturb the stability and convergence of the total method need to be considered. This section will consider these points and develop the order changing scheme used in D4.

A concise definition of the stepsize scheme containing the stepsize and order selection algorithms will depend on several definitions, given

below.

Definition 3.1

A step selection scheme is a function θ such that

$$h_n = h\ \theta(t_n,h) \tag{3.17}$$

where, for some $h > 0$, $t_0 \leq t \leq t_f$,

$$1 \geq \theta(t,h) \geq \Delta > 0.$$

Definition 3.2

A formula selection scheme is a function $I(h,t)$ such that the formula used for the interval $[t_n, t_{n+1}]$ is $F_{I(h,t_n)}$ where h is the parameter of definition 3.1.

In the following, the formulas F_i are the 11 formulas (2.12) and F_i is the formula of order i. Thus, $I(h,t_n)$ is an order selection scheme for D4.

Definition 3.3

A method is stable with respect to a step selection scheme θ and a formula selection scheme I if there exists a constant $M < \infty$, dependent only on the differential equation, such that

$$|y_m - \tilde{y}_m| < M\ |y_n - \tilde{y}_n| \tag{3.18}$$

for all $t_0 \leq t_n < t_m \leq t_f$, where y_i and $\tilde{y}_i$ are two numerical solutions to the same problem using the same variable step, variable order method.

Definition 3.4

A set of formulas $\{F_i\}$ is convergent with respect to θ and I if the computed solution y_n converges to $y(t_n)$ for any $t_0 \leq t_n \leq t_f$ as $h \to 0$ and the starting errors $\to 0$.

Gear and Tu [22] and Skeel [39] investigate the effect of varying the stepsize on the stability and convergence of normal form methods, i.e. methods which store the dependent variables in Nordsieck form, and derive stepsize changing criteria which result in convergence of such methods with respect to stepsize changing schemes θ which meet such criteria. Gear and Tu also consider variable step Adams methods that store $\underline{y}_n = [y_n, hy'_n, \ldots, hy'_{n-k}]$ and use stepsize dependent coefficients $\alpha^{(0)}_{i,n}$, $\alpha^{(1)}_{i,n}$. These methods are compared with the normal form methods and are shown to have less stringent requirements on the stepsize selection scheme.

Gear and Watanabe [23] consider the effect of varying the order of normal form methods that are stable and convergent with respect to some stepsize scheme θ. Criteria for the order selection scheme I are established which result in stability and convergence of normal form methods with respect to I.

The basic construct of these investigations is the stability matrix S_n of section 2.3 based on the matrix representation of the iterated corrector formula (3.5) with the extrapolating predictor P. For normal form methods, this is

$$S_n = \prod_{m=1}^{M_n} [I - \sum_{j=1}^{s} (\underline{\ell}_j \underline{e}_j^T - h_n f_y^{(j-1)} \underline{\ell}_j \underline{e}_0^T)]P, \qquad (3.19)$$

where $f_y^{(i)}$ are the Jacobian matrices of the derivatives and P is Pascal's triangle matrix.

The stepsize changing algorithm can be represented as

$$C_n = \text{diag}\,[1, \alpha, \ldots, \alpha^q]$$

for $\alpha = h_n/h_{n-1}$ when changing from the stepsize h_{n-1} to the new stepsize

h_n. A number of possibilities exist for changing the order. Shampine and Gordon (1973) have calculated the exact procedure when using the Nordsieck representation of the dependent variables to achieve the same result as when a q-th order Adams method using the representation

$$\underline{y}_n = (\underline{y}_n, hy_n', \ldots, hy'_{n-k})$$

changes from order q to either q-1 or q+1. Brown [6] develops this for Gear's backward differentiation formulas. However, these are quite complicated and it happens that the following simpler device will suffice. When the order is lowered or remains the same, the value $(\underline{a}_n)_q$ is dropped or left, respectively; when the order is raised, a new value $(\underline{a}_n)_{q+1}$ is calculated by differencing $((a_n)_q - (a_{n-1})_q)/(q+1)$. Alternatively, this term could be left alone and the initial estimate would be $(\underline{a}_n)_{q+1} = 0$. If $O_{i,j}$ is the matrix representation of the order changing algorithm for going from order i to j, then

$$O_{q,q} = I, \tag{3.20}$$

$$O_{q,q-1} = I^q = I - (\underline{e}_q \, \underline{e}_q^T)$$

$$O_{q,q+1} = \begin{pmatrix} I & & 0 \\ & \Delta_n & \\ 0 & & I \end{pmatrix} \begin{matrix} \text{(0) row} \\ \\ \} \text{ row } q+1 \\ \\ \\ \underbrace{\qquad}_{\text{column } q} \end{matrix}$$

where $\Delta_n = \frac{1}{q+1}\left(1 - \frac{y_{n-2}^{(q)}}{y_{n-1}^{(q)}}\right)$. Alternatively,

$$0_{q,q+1} = I^{q+1}.$$

The method can then be described by

$$\underline{a}_n = S_n\, C_n 0_n\, \underline{a}_{n-1}, \tag{3.21}$$

$$\underline{a}_n = T_n\, \underline{a}_{n-1}.$$

If we define

$$S_n = \hat{S}_q + h_n\, \tilde{S}_n, \tag{3.22}$$

$$\hat{S}_q = (I - \sum_{j=1}^{s} \underline{\ell}_j\, \underline{e}_j^T) M_n P,$$

where S_q corresponds to the order q formula in (2.12), then the eigenvalues of

$$\hat{T}_n = \hat{S}_q\, C_n\, 0_n \tag{3.23}$$

can be used to determine if the method will be stable and convergent since if the eigenvalues of $\hat{T}_n$ exceed 1 in norm then even for small h_n the computation will not converge.

These matrices have been used to establish criteria for stepsize changes which allow a predictor-corrector method of form (3.9) to be convergent and stable with respect to the stepsize changing scheme $\theta(t_n, h)$ where

$$h_n = \theta(t_n, h)\, h,$$

$$h \geq \max (h_i),$$

$$1 \leq i \leq N$$

for N the number of steps taken in integrating from t_0 to t_f. The relevant theorems from Skeel [39] are presented in

Lemma 3.1

If a variable step method is such that

1) The underlying formula F_q is strongly stable;

2) F_q has order $q \geq 1$;

3) There exist positive constants Δ and V such that

$$0 < \Delta \leq \theta_n \leq 1$$

and

$$\sum_{n=1}^{N} | \theta_n - \theta_{n-1} | \leq V;$$

then the method is stable and convergent with respect to θ.

Gear and Watanabe sought a criterion to limit the frequency of order changes that would allow a multivalue method to be stable and convergent with respect to a formula selection scheme. The result of interest may be expressed as

Lemma 3.2

If a variable step, variable formula method is such that

1) Each formula F_q is strongly stable;

2) Each formula has order $q \geq 1$;

3) The order changing operations are exact for $y' = 0$, and $|| O_{i,j} ||$ is bounded;

4) The coefficients in the method M_q corresponding to F_q are bounded;

5) Each formula is stable and convergent with respect to its stepsize changing scheme;

then, for each pair of formulas F_i, F_j, there exists an integer p_{ij} such that the method is stable and convergent with respect to a formula selection scheme that produces infrequent formula changes in the sense that at least p_{ij} steps are taken without formula change after formula

F_i has been selected before formula F_j is selected.

The values of p_{ij} of interest in D4 are listed in Table 3.1.

i \ j	1	2	3	4	5	6	7	8	9	10	11
1	1	2									
2	1	1	3								
3		1	1	2							
4			1	1	2						
5				1	1	4					
6					1	1	2				
7						1	1	3			
8							1	1	9		
9								1	1	4	
10									1	1	3
11										1	1

Table 3.1. $p_{i,j}$ for changing order in D4.

There is now enough information about the behavior of variable step, variable order normal form methods to make a statement about their stability and behavior with respect to changing stepsize. In particular, the method M_{D4} as presented in section 3.2 and this section can be shown to have desirable stability properties. This can be seen by considering the use of M_{D4} on a problem

$$y' = f(y,t),$$

$$y(t_0) = y_0,$$

$$t_0 \leq t \leq t_f,$$

where $f^{(r)}(y,t)$ is continuously differentiable with respect to t and $f_y^{(j)}(y,t)$, $j = 1,\ldots,s-1$, is continuous in y for $t_0 \leq t \leq t_f$ and bounded

y. In particular, if r is the highest order formula used, s = 1 for $r \leq 2$, s = 2 for $r \leq 5$, s = 3 for $r \leq 7$, and s = 4 for $r \leq 11$. The restraints on f(y,t) satisfy the hypotheses of theorem 2.2 so the asymptotic error estimate will be applicable. M_{D4} also requires that a minimum allowable stepsize H_{min} be chosen to keep the error calculation from being affected by machine roundoff, and that a maximum useable stepsize H_{max} no greater than the interval of integration be imposed. Hull and Enright [29] discuss the automatic choice of these values for different integrators. However, in D4 the choice is left to the user. The method implemented in the program D4 meets the hypotheses of Lemma 3.1 with the parameters

$$h \leq H_{max},$$

$$h_n = \theta(t_n,h)h,$$

$$\Delta = H_{min}/h,$$

$$N \leq \tilde{N} = (t_f - t_0)/H_{min},$$

$$V \leq \sum_{n=1}^{N} | 1 - \Delta | \quad .$$

Further, the hypotheses of Lemma 3.2 are met if the order is never increased after less than q+1 steps, for q the current order, since $p_{q,q+1} \leq q+1$ for all values in Table 3.1. This is true in the program D4.

Hence, the method M_{D4} is stable with respect to the stepsize and order changing schemes. Since a lower bound is placed on H, the stepsize, the method cannot converge to the true solution as h, or H_{max}, is forced to zero. However, if greater accuracy is requested on one machine integration than on another, greater accuracy can be expected. Note that the method is not guaranteed to integrate to $y(t_f)$; the corrector might not

converge or the calculated error per step may not be small enough for some $h_n = H_{min}$. Also, if $y(t_f)$ is computed, the global error has not been estimated, although at each step the relative error per step was near EPS, a user input.

3.5 Higher Derivative Calculation

The most significant drawback to using D4 is the requirement that derivatives of the first order equation solution be available through order 4. A permanent disadvantage is that the calculation of each added derivative will increase the computation time as least linearly with S, probably more, for any function of interest. Further, the average user does not want to have to program these added derivatives or even calculate what they are, since this can be quite time consuming and tedious. Several means of alleviating this second problem are available.

If the system of equations is autonomous and the exact Jacobian $f_y(y_n)$ is available through a subroutine call, then

$$f'(y) = f_y(y_n)\, f(y_n) \tag{3.24}$$

which can be used for up to fifth order A-stable formulas in D4. If the Jacobian is easy to calculate because the system is not highly cross coupled, or the Jacobian is well known, or if the system has a linear dependence on y as in

$$\underline{y}' = A\underline{y} + f(t), \tag{3.25}$$

and f(t) is easily differentiable, then a similar device can be used. Enright [16] provided (3.24) as the justification for his use of second derivative methods that are A-stable to 4th order. Also, for linear and simply coupled autonomous systems it may be easy to calculate

$$f'_y = f_{yy} f + f_y^2,$$

which is used in the convergence matrix W.

If the function f can be given as an algebraic expression of a limited number of functions, then various algebraic manipulation languages can be used to compute the necessary derivatives; f_y, f'_y, ..., may also be computed.

By using I/O facilities in these languages an entire FORTRAN subroutine DIFFUN could be produced.

FORMAC, a PL/I precompiler described by Fike [17], has a function DERIV(F,T,N) that finds $\frac{d^N F}{dT^N}$ where F is an algebraic function of T and any of (SIN, COS, SINH, COSH, ATAN, ATANH, LOG, LOG10, LOG2, ERF).

ALTRAN is a FORTRAN based language that handles ratios of polynomials, as well as real numbers, integers, and rational numbers. The functions DIFFN(F,T) in ALTRAN serve the same function as DERIV in FORMAC. The function TPS(F, T, N) calculates the Truncated Power Series (Taylor series) of F up to order N.

A common method for solving differential equations is to calculate the Taylor series at a point (y_n, t_n) and then find (y_{n+1}, t_{n+1}) using specified error criteria to choose $h_{n+1} = t_{n+1} - t_n$. One program that does this is called TAYLOR, which is written in BCPL and which accepts a description of the system involving expressions in any of the FORTRAN functions and outputs two FORTRAN subroutines.

One of these output subroutines (the name is given by the user) accepts initial values and calculates the first Taylor series; the other finds a value at a given t by taking as many steps as necessary. Two additional routines, XTAY01 and XTAY02, are also produced and are called

by the first two.

The method used was coded by Norman [36], who describes the program TAYLOR, and Barton et al [1] derived the underlying formulas. The elements of the Taylor series with h = 1 are calculated using recurrence relations based on the simplest rules of differentiation. For example, if

$$y' = y_1 y_2,$$

$$y_2' = y_2,$$

then

$$y_1^{(i)}/i! = (i-2)!(y^{(i-1)} y_2^{(i-2)} + y^{(i-2)} y_2^{(i-1)})/i,$$

$$y_2^{(i)}/i! = y_2^{(i-1)}/i.$$

can be solved for i = 2,...,s. See Barton et al [1] for a more detailed discussion.

Since the TAYLOR system was used to provide the up to 4th derivative calculator DIFFUN in the test of D4 in section 3.6, the special use of TAYLOR for that purpose is described below for a simple example.

The input program for y' = y would be as in Figure 5.

```
COMMENT     Y' = Y
TERMS       4
INITIAL     SET (T, Y, TERMS)
ADVANCE     VAL (T, Y)
EQUATIONS
Y' = Y
END
```

Figure 5. Input to TAYLOR to solve y' = y.

The output would be a FORTRAN program SET (T, Y, S) which, given T and Y(T), produces the 0-th through s-th terms of the Taylor series and calls XTAY02 while doing so. SET is changed slightly to become DIFFUN (T,H,Y,DYS,S). VAL would be used to find Y(T) by the Taylor method and TAYLOR will not compile unless some minimal routine is required. TERMS 4 limits the terms in the series to 0 through 4 and saves storage by doing so.

3.6 Numerical Tests

Ehle [15] has tested a number of FORTRAN programs designed for solving stiff systems of ordinary differential equations using three values of Σ on each of 24 small systems ($N \leq 9$). Each problem is specified by

$$p = \langle f, t_0, y_0, t_f, S, YMAX_0, \varepsilon, h_{max}, h_{min}, \text{code} \rangle$$

for

$$y' = f(t,y),$$

$$y(t_0) = y_0$$

and $YMAX_0$ is the initial value of y_{max} for a relative error calculation in which the truncation error per step is calculated relative to $\max(y_{max}, y_n)$, t_f is the final value of t, S is a set of values of t at which the global error is to be calculated, and the code specifies various parameters of the problem, such as a stiffness ratio, the linearity of the problem, and the range of α such that all eigenvalues are in $S(\alpha)$ as defined in Chapter 1. This last is the most important for D4 since in general none of the methods in Ehle's tests did as well for $\alpha \geq 30°$ as for $\alpha < 30°$. Since D4 is never restricted by the eigenvalues as long as they have negative real parts, tests were made which show that D4 did as well for $60° \geq \alpha > 30°$ and $90° \geq \alpha > 60°$ as for $30° \geq \alpha > 0°$, in the sense that no serious convergence problems or large errors occurred during the calculations. The global error

at points in S were monitored for those tests having closed solutions, and these were usually within a factor of 10 of EPS, the requested error per step.

A subset of Ehle's problems was chosen to include as many problems with $\alpha > 30^{\circ}$ as possible as well as some simpler benchmark tests, and these were compiled using TAYLOR. Some tests were eliminated that were inappropriate, e.g., one linear constant coefficient problem had very large values in its corresponding matrix, and since TAYLOR subroutines calculate the derivatives without scaling by H the higher derivatives produced overflow errors.

The number of calls to DIFFUN by D4 and the number of matrix inversions for W were measured. W was always calculated by numerical differencing but this use of DIFFUN was not included in the recorded number of calls by D4. Results are in Table 3.2 and the test problems are listed in Appendix II. NFNS is the number of calls to DIFFUN, and NW the number of matrix inversions.

TEST	eps	D4 NFNS	D4 NW	DIFSUB NFNS	DIFSUB NW
A	10^{-3}	285	91	122	12
$\alpha = 0°$	10^{-6}	775	207	263	19
	10^{-9}	3186	807	380	17
B		79	12	124	12
$\alpha = 0°$		188	25	244	17
		433	40	333	17
C		864	76	93	11
$30° < \alpha < 60°$		210	49	229	29
		311	56	249	19
D		126	22	150	14
$\alpha = 0°$		346	60	354	30
		1204	202	405	22
E		136	23	173	18
$30° < \alpha < 60°$		329	43	351	27
		1434	185	538	21
F		490	34	1752	9
$60° < \alpha$		1709	38	2181	16
		5631	48	3213	11
G		375	23	3577	18
$60° < \alpha$		1251	33	5471	8
		3115	47	5354	14
H		447	30	1147	8
$60° < \alpha$		1737	46	2233	11
		5621	57	3006	11
I		37	10	52	9
$\alpha = 45°$		28	6	48	6
		25	6	38	5
J		26	12	21	2
$\alpha = 45°$		26	12	21	2
		26	12	21	2
K		322	41	52	9
$60° < \alpha$		318	47	426	31
		727	111	421	25
L		75	25	61	20
$60° < \alpha$		75	25	71	22
		61	24	68	24
M		412	56	47	9
$60° < \alpha$		142	26	108	18
		659	103	479	36

Table 3.2. Comparison of D4 and DIFSUB on 13 tests from Ehle [15].

4. Alternatives and Conclusions

The value of multi-derivative methods lies in their applicability to a wide variety of stiff problems, many of which cannot be properly solved by the less expensive to use first derivative methods such as Runge-Kutta, Adams, and backward differentiation. By using A-stable higher derivative methods, as large a stepsize as is consistent with the error criterion can be taken at any order given suitable start up and order changing procedures. In this chapter, methods that require only the first derivative yet retain some of the properties of the higher derivative methods are presented. The final section considers the usefulness of the methods presented throughout.

4.1 Matrix-Valued Coefficient Methods

When higher derivative formulas cannot be used, methods which share some of their desirable properties can be substituted; in particular, if we consider that

$$(f_y)^j y \doteq (f_y)^{j-1} y' \doteq y^{(j)}, \tag{4.1}$$

and in the linear constant coefficient case, $y' = Ay$, (4.1) is exact, then formulas can be developed which use only y_n and y_n' but have coefficients that include powers of an approximation Q_n to the Jacobian $f_y(y_n,t_n)$. Since the approximations (4.1) are not exact for non-linear problems even if the calculated f_y is exact, additional non-zero coefficients must be added to insure that the order is not lowered. Since these new non-zero coefficients may destroy the property of A-stability of a formula, new non-zero coefficients will have to be added to restore A-stability or at least stiff stability. Lambert and Sigurdsson [31] developed formulas

$$0 = \sum_{i=0}^{k} [a_i^{(0)} I + \sum_{j=1}^{s} a_i^{(j)} h^j Q_n^j] y_{n-j} + \tag{4.2}$$
$$h \sum_{i=0}^{k} [b_i^{(0)} I + \sum_{j=1}^{s-1} b_i^{(j)} h^j Q^j] f_{n-j}.$$

By forming a Taylor expansion around $y(t_n)$, a method of a given order p can be obtained by matching terms in $h^i Q_n^j y^{i-j}$ for $i = 0,\ldots,p$, for each $j = 0,\ldots,s$. This leads to a linear system of $p+1-j$ equations for each value $j \geq 0,\ldots,s$.

If $Q_n = \lambda$ for the test equation

$$y' = \lambda y,$$

then the stability polynomial becomes

$$\rho(\xi) + \sum_{j=1}^{s} (h\lambda)^j \hat{\sigma}_j(\xi) = 0 \tag{4.3}$$

for

$$\hat{\sigma}_j(\xi) = \sum_{i=0}^{k} (a_i^{(j)} + b_i^{(j-1)})\xi^{k-i}, \tag{4.4}$$

$$\rho(\xi) = \sum_{i=0}^{k} a_i^{(0)} \xi^{k-i}.$$

If the additional coefficients available as $a_i^{(j)}$, $b_i^{(j)}$, $j > 0$, beyond the $p+1-j$ required to establish the order of the method are used to improve the stability of the method, then the extra matrix computations required by a method based on a formula (4.1) are justified. The stability region depends on terms $(h\lambda)^j \hat{\sigma}_j$ and

$$\hat{\sigma}_j = (a_0^{(j)} + b_0^{(j-1)})(\xi - \hat{\xi}_j) \prod_{\ell=2}^{k} (\xi - \xi_\ell), \tag{4.5}$$

where some arbitrary $\hat{\xi}_j$ will be called the "root of interest" and the ξ_ℓ, $\ell = 2,\ldots,k$ are "extraneous roots" of $\hat{\sigma}_j$. Define $\pi = \prod_{\ell=2}^{k} (\xi - \xi_\ell)$ for further use. If the extraneous roots of $\hat{\sigma}_j$ were made the same as the extraneous roots of $\rho(\xi)$ then for these roots $|\xi_\ell| < 1$ for all values of $h\lambda$ in the stability polynomial (4.3). Thus, the stability region depends only on the principal root 1 of ρ and the roots of interest of the $\hat{\sigma}_j$.

The above can be accomplished by setting

$$\sum_{i=0}^{k} (a_i^{(j)} + b_i^{(j-1)})\, \xi_q^{k-i} = 0, \qquad (4.6)$$

$$q = 2,\ldots,k,$$

for ξ_q a root of $\rho(\xi)$, $\xi_1 = 1$. This adds an extra k-1 constraints to a linear system to be solved for coefficients

$$a_i^{(j)},\ b_i^{(j)},\ i = 0,\ldots,k$$

for each value $j = 1,\ldots,s$. Lambert and Sigurdsson call a formula that satisfies (4.6) stabilized.

The form of formula (4.1) requires that $b_i^{(s)} = 0$ for all i, so the minimum value of s for a stabilized formula of a given order p can be found by requiring that the k+1 coefficients $a_i^{(s)}$, $i = 0,\ldots,k$, have no more than k+1 constraints on them. There are p+1-s constraints to achieve order p, and k-1 stabilizing constraints if $a_i^{(s-1)} \neq 0$, $i = 0,\ldots,k$. Thus,

$$p + k - s \leq k + 1;$$

$$s \geq p - 1.$$

Since up to 2k+2 values $a_i^{(j)}$, $b_i^{(j)}$, can be non-zero, these values can be picked to produce formulas (4.2) that are A-stable for $Q_n = f_y$. In particular, if the coefficients are chosen in terms of parameters such that the order and stabilizing conditions are met for all values of the parameters, then the parameter values for which the formula is A-stable for $Q_n = f_y$ can then be determined and, within those constraints, particular values may be chosen. One may choose the parameters to minimize the truncation error coefficients for accurate approximations Q_n to f_y, or to minimize the dependence of the truncation errors on Q_n, or to optimize the error in some other way.

An example of this from Lambert and Sigurdsson [31] is the third order formula

$$0 = [-1 + (\tfrac{1}{3}+2a)Q_n - aQ_n^2]y_n + [(1+a) + \tfrac{1}{12}(-15+\alpha+\beta-2\beta a(1+\alpha))Q_n \qquad (4.7)$$

$$-bQ_n^2]y_{n-1} + [-(\alpha+\beta) + \tfrac{1}{3}(4-\beta+6a(\alpha+\beta)Q_n + (3a+2b)Q_n^2]y_{n-2} +$$

$$[\beta + \tfrac{1}{12}(-5-\alpha+3\beta-24\alpha\beta)Q_n + (2a+b)Q_n^2]y_{n-3} +$$

$$[\tfrac{1}{12}(23-5\alpha-\beta) + (\tfrac{1}{6}-a(1-\alpha)+b)Q_n]f_{n-1} + [\tfrac{-2}{3}(2+\alpha-\beta) -$$

$$(\tfrac{1}{6}\alpha + a(3-\alpha+\beta) + 2b)Q_n]f_{n-2} + [\tfrac{1}{12}(5+\alpha+5\beta) + (\tfrac{1}{6}\beta+a(2-\beta)+b)Q_n]f_{n-3},$$

for $\alpha = \xi_2 + \xi_3$, $\beta = \xi_2\xi_3$, $|\xi_i| < 1$, $i = 2,3$, $a \geq \frac{1}{12}$.

The truncation error is

$$h^4[\tfrac{1}{24}(9+\alpha+\beta)y^{(4)} + [\tfrac{1}{36}(14+\alpha+2\beta) + \tfrac{1}{6}a(17+\alpha-\beta)]Q_n y^{(3)} +$$

$$(3a+b)Q_n^2\, y^{(2)}] + O(h^5).$$

The stability region for (4.7) for the parameter values

$$\alpha = \beta = 0,$$

$$a = 1/6,\ b = -1/2$$

is plotted in Figure 6.

The extraneous roots of the characteristic subpolynomials $\rho(\xi)$, $\hat{\sigma}_1(\xi)$, and $\hat{\sigma}_2(\xi)$ for the above method are the ξ_2, ξ_3 used in choosing α and β. For $\xi_2 = \xi_3 = 0$, the characteristic polynomial becomes

$$\xi^2[(\xi-1) + h\lambda(\xi+1/2) + h^2\lambda^2\xi] = 0. \qquad (4.8)$$

Note that if the approximation Q_n to f_y is not exact then the stability region $R(\varepsilon)$ for $Q_n = (\lambda+\varepsilon)$, $y' = \lambda y$, is all $h\lambda$ which satisfy the root condition for

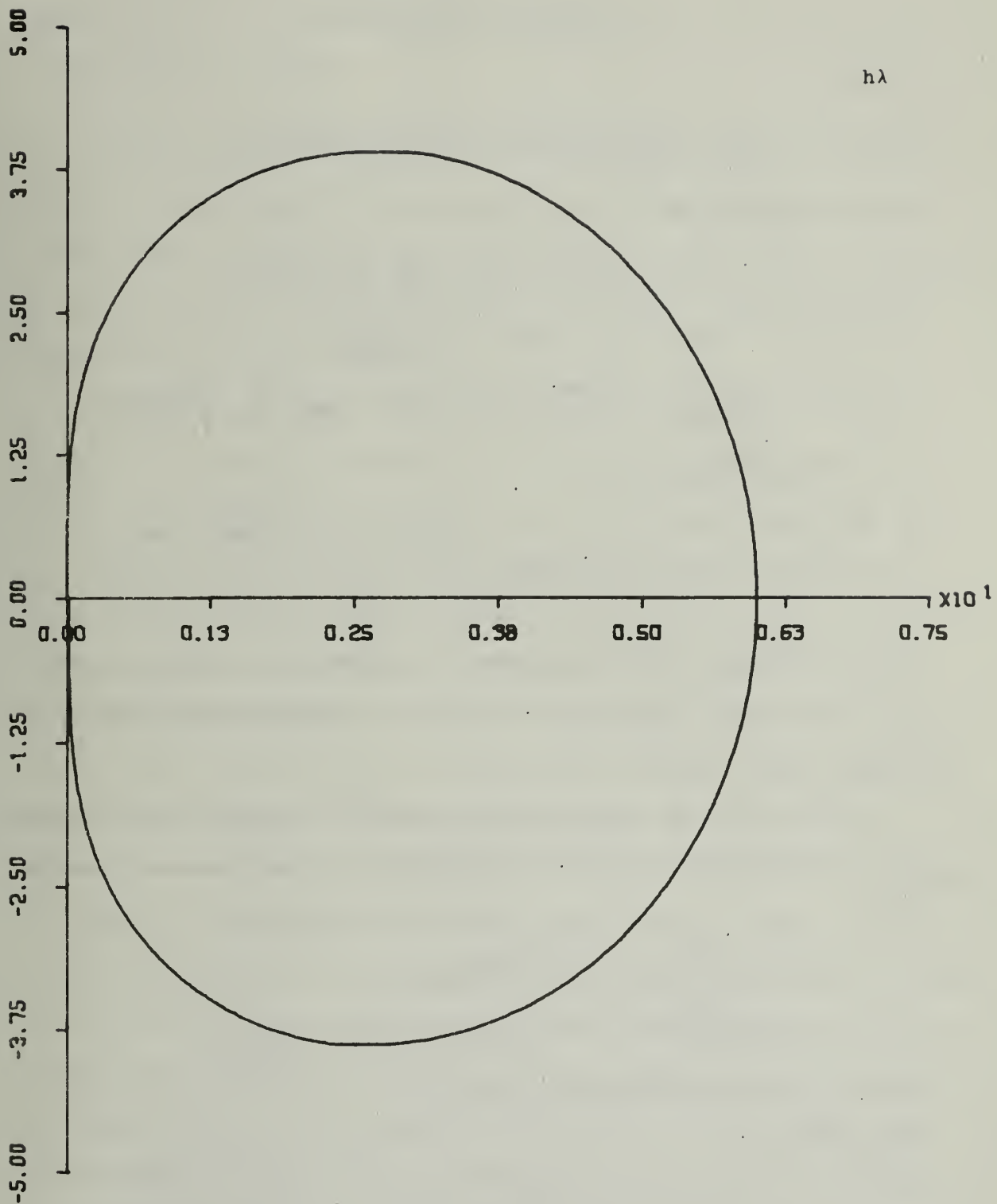

Figure 6. Stability region when $Q_n = \lambda$ for matrix valued coefficient formula (4.7).

$$\rho(\xi) + h\tilde{\sigma}_1(\xi) + h^2\tilde{\sigma}_2(\xi) = 0, \tag{4.9}$$

where

$$\tilde{\sigma}_j(\xi) = \sum_{i=0}^{k}[a_i^{(j)}(\lambda+\varepsilon)^j + b_i^{(j-1)}(\lambda+\varepsilon)^{j-1}\lambda]\xi^{k-i}.$$

This is the same as

$$\tilde{\sigma}_j(\xi) = Q_n^j[\hat{\sigma}_j(\xi) - Q_n^{-1}\varepsilon \sum_{i=0}^{k} b_i^{(j-1)}\xi^{k-i}].$$

Then for π defined as for (4.5), (4.9) becomes

$$\pi[(\xi-1) - \frac{h\varepsilon}{\pi}(\sum_{i=0}^{k} b_i^{(0)}\xi^{k-i}) + hQ_n[(\xi-\hat{\xi}_1) - \frac{h\varepsilon}{\pi}(\sum_{i=0}^{k} b_i^{(1)}\xi^{k-i})]$$

$$+ h^2Q_n^2(\xi-\hat{\xi}_2)] = 0. \tag{4.10}$$

In the example above, $\hat{\xi}_1 = -1/2$, and $\hat{\xi}_2 = 0$. For h and ε small enough and ξ near the unit circle, the $\frac{h\varepsilon}{\pi}$ terms are minor perturbations to all but the high order term of a quadratic equation in $h\lambda$. The perturbed stability region $R(\varepsilon)$ is thus very close to the unperturbed region R, and converges to it as $\varepsilon \to 0$.

If a formula of form (4.2) were used in an iterated corrector method with a quasi-Newton corrector, the corrector calculation would have the form

$$(I - ha_0^{(1)}Q_n - h^2a_0^{(2)}Q_n^2)(\underline{y}_{n,(m)}) \tag{4.11}$$

$$= -h[\frac{dG}{d\underline{y}}]^{-1}(b_0^{(0)} + b_0^{(1)}Q_nh)(\underline{f}_{n,(m+1)} - \underline{f}_{n,(m)}),$$

where G is the difference function (4.2).

Thus, if $Q_n \doteq f_y$,

$$\frac{dG}{dy} \doteq -I + (a_0^{(1)} + b_0^{(0)})hQ_n + (a_0^{(2)} + b_0^{(1)})h^2Q_n^2. \tag{4.12}$$

Since the inverse of $\frac{dG}{dy}$ is already needed, premultiplying $\frac{dG}{dy}$ by

$$(I - ha_0^{(1)}Q_n - h^2 a_0^{(2)}Q_n^2)$$

before inverting yields

$$\underline{y}_{n,(m+1)} - \underline{y}_{n,(m)} = \tilde{W}\, h(b_0^{(0)} + hb_0^{(1)}Q_n)(\underline{f}_{n,(m+1)} - \underline{f}_{n,(m)}), \quad (4.13)$$

where $\tilde{W}$ is the inverse of a linear combination of hQ_n terms up to the fourth power. By modifying an existing program such as DIFSUB to compute $\tilde{W}$ instead of W, and premultiplying $\Delta f_{n,(m)}$ by $h(b_0^{(0)} + hb_0^{(1)}Q_n)$, these methods can be easily implemented.

Since the local truncation error is known in terms of $y^{(4)}$, $Q_n y^{(3)}$, and $Q_n^2 y^{(2)}$ this can be computed directly when the Nordsieck form is used to store the independent variables.

Although a third order A-stable formula has been developed, the computational expense of this method relative to a third order backward differentiation formula that is stiffly stable, or a second derivative A-stable method, is large. The formula is only guaranteed to be A-stable for $Q_n = f_y$; if Q_n is in error the stability region may be worse than for the third order backward differentiation method. Also, the solution of a linear system of up to the fourth power of hQ_n is necessary at each step. Even when an LU decomposition and backward substitution scheme is used, this is more expensive than for the multiderivative formula which requires the solution of a linear system with only second order terms in hQ_n, thus saving on the computation to generate the third and fourth powers of Q_n. Also, since $(\underline{f}_{n,(m+1)} - \underline{f}_{n,(m)})$ must be multiplied by $(b_0^{(0)} + hb_0^{(1)}Q_n)$ at every corrector iteration, the corrector may be as expensive as forming the second derivative, and in the linear constant coefficient case it is. The error estimator also involves multiplication of derivatives by Q_n and

Q_n^2, making the error estimator more expensive. However, if one assumes

$$y^{(4)} = Q_n y^{(3)} = Q_n^2 y^{(2)} \tag{4.14}$$

then the error can be estimated in terms of $y^{(4)}$ only. However, a serious error may be committed when (4.14) is not true.

Thus, while matrix valued coefficient methods can give A-stable methods when the matrices are accurate approximations to the Jacobian matrix, they can be as expensive to use as corresponding multiderivative methods and yet have inferior stability properties when the approximation to the Jacobian loses accuracy.

4.2 Conclusions

A set of formulas for the numerical integration of systems of ordinary differential equations has been presented all of whose elements are A-stable, stable at infinity, and convergent for order up to 11. Therefore, they are useful in solving stiff problems, especially those for which the eigenvalues of the Jacobian matrix of the system at some point (y_n, t_n) have large imaginary parts. In return for the high order of these A-stable methods, derivatives higher than the first are required to solve first order systems of equations. It must be left to a potential user of the computer program D4 to decide whether this tradeoff is feasible; however, several convenient methods for providing the higher derivatives were presented and should lighten the burden on any potential user.

An alternative approach, previously analyzed by Lambert and Sigurdsson [31], was presented in the matrix valued coefficient formulas. However, it was noted that even the simplest such matrix valued coefficient formula provides a method which requires a large amount of work in the form of matrix evaluations and manipulations. Further, the methods are only

guaranteed to be A-stable for an exact Jacobian and are not necessarily stable at infinity.

The program listed in Appendix I has been tested with subroutines DIFFUN based on the tests in Appendix II. The results were indicative of the power of the underlying method, but since no "fine tuning" on an extensive set of problems over a substantial period of time was undertaken, it is appropriate to list here possibilities that might improve the performance of D4, either by decreasing overhead, computation time, or number of function evaluations.

1. Matrix inversion - D4 solves linear systems of equations using the LU decomposition and back substitution subroutines DECOMP and SOLVE from Moler [33]. If the Jacobian matrix is known to be sparse, or tri-diagonal, or have some other property for which a more effective algorithm exists, then an individual user may substitute these.

2. Corrector Iteration - Hindmarsh [27] suggests that the user be allowed four choices concerning the corrector iteration. He can allow the Jacobian to be approximated by numerical differencing, or provide it exactly. In either case, a Newton-type iteration is used. He can also approximate the Jacobian with a diagonal matrix, in which case the iteration becomes a chord method. This saves time computing PW, but may increase the number of calls to DIFFUN. Or he may use straight functional iteration, so no computation time is required to find PW; however, the corrector will only converge for small stepsize h.

3. Norms - The norm in which the error is calculated and in which the corrector convergence test is applied can be modified. Currently, the error estimate E_n is computed with the Euclidean norm, but the max norm is

used to decide if the corrector has converged. If the Euclidean norm is used instead, requiring more computation, a look ahead feature in which the corrector value at the next iteration is "predicted" by multiplying by a previous ratio of two consecutive correctors can be used. If it is "predicted" that the next correction will be much smaller than the convergence criterion, then that correction will not be done. Such a change to DIFSUB resulted in a 10% reduction in calls to DIFFUN on a set of test equations.

4. Heuristic tuning - numerous bookkeeping operations involve constants that were picked heuristically. An example is the values 1.2 and 1.3 used to divide next stepsizes in the order and stepsize selection algorithm. Also rules for deciding how often to update PW and how to control for repeated failures to meet the error criterion or for failure of the corrector to converge, were all picked heuristically on the basis of experience in using DIFSUB. Further fine tuning on D4 is also possible.

However, even without this additional work of fine tuning the program D4 can be used to solve very stiff systems of ordinary differential equations using large stepsizes due to the high order available in its A-stable formulas.

LIST OF REFERENCES

[1] Barton, D., Willers, I. M., and Zahar, R. V. M., "Taylor Series Methods for Ordinary Differential Equations--An Evaluation," Mathematical Software, Academic Press, New York, 1971.

[2] Bashforth, F. and Adams, J. C., Theories of Capillary Action, Cambridge U.P., New York, 1883.

[3] Bjurel, G., Dahlquist, G., Lindberg, B., Linde, S., and Odén, L., "Survey of Stiff Ordinary Differential Equations," Dept. of Information Processing, Royal Institute of Technology, Stockholm, Report #NA70.11, 1970.

[4] Blue, J. L., and Gummel, H. K., "Rational Approximations to Matrix Exponential for Systems of Stiff Equations," J. Comp. Physics, 5, pp. 70-83, 1970.

[5] Brown, R. L., "Second Derivative Methods for the Solution of Stiff Ordinary Differential Equations," Dept. of Computer Science, University of Illinois at Urbana-Champaign, File No. 877, 1973.

[6] Brown, R. L., "Recursive Calculation of Corrector Coefficients," ACM SIGNUM Newsletter, 8, No. 4, 1973.

[7] Brown, W. S., ALTRAN User's Manual, Bell Laboratories, Murray Hill, New Jersey, 1973.

[8] Cavendish, J. C., Culham, W. E., and Varga, R. S., "A Comparison of Crank-Nicolson and Chebyshev Rational Methods for Numerically Solving Linear Parabolic Equations," J. Comp. Physics, 10, pp. 354-368, 1972.

[9] Coppel, W. A., Stability and Asymptotic Behavior of Differential Equations, D. C. Heath, Boston, Massachusetts, 1965.

[10] Curtiss, C. F., and Hirschfelder, J. O., "Integration of Stiff Equations," Proc. Nat. Acad. Science, U.S., 38, pp. 235-243, 1952.

[11] Dahlquist, G., "Numerical Integration of Ordinary Differential Equations," Math. Scandinavica, 4, pp. 33-50, 1956.

[12] Dahlquist, G., "Stability and Error Bounds in the Numerical Integration of Ordinary Differential Equations," Trans. Roy. Inst. Tech., Stockholm, No. 130, 1959.

[13] Dahlquist, G., "A Special Stability Problem for Linear Multistep Methods," BIT, 3, pp. 27-43, 1963.

[14] Ehle, B., "High Order A-Stable Methods for the Numerical Solution of Systems of Differential Equations," BIT, 8, pp. 276-278, 1968.

[15] Ehle, B., "A Comparison of Numerical Methods for Solving Certain Stiff Ordinary Differential Equations," Dept of Mathematics, University of Victoria, Victoria, Report #70, 1972.

[16] Enright, W., "Studies in the Numerical Solution of Stiff Ordinary Differential Equations," Dept. of Computer Science, University of Toronto, Toronto, Report #46, 1972.

[17] Fike, C. T., PL/I for Scientific Programmers, Prentice-Hall, Englewood Cliffs, New Jersey, 1970.

[18] Gear, C. W., "The Automatic Integration of Stiff Ordinary Differential Equations," Information Processing, 68, ed. A. J. H. Morrel, North Holland Publishing Company, Amsterdam, pp. 187-193, 1969.

[19] Gear, C. W., Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[20] Gear, C. W., "The Automatic Integration of Ordinary Differential Equations," Comm. ACM 14, pp. 176-179, 1971.

[21] Gear, C. W., "DIFSUB for Solution to Ordinary Differential Equations," Comm. ACM, 14, pp. 185-190, 1971.

[22] Gear, C. W., and Tu, K. W., "The Effect of Variable Mesh Size on the Stability of Multistep Methods", submitted to SIAM J. Num. An., 1973.

[23] Gear, C. W., and Watanabe, D. S., "Stability and Convergence of Variable Order Multistep Methods," submitted to SIAM J. Num. An., 1973.

[24] Genin, Y., "An Algebraic Approach to A-Stable Linear Multistep-Multi-derivative Integration Formulas," MBLE Research Laboratory Report R244, 1974.

[25] Graves-Morris, P.R. ed, Padé Approximants and Their Applications, Academic Press, London, 1973.

[26] Henrici, P., Discrete Variable Methods for Ordinary Differential Equations, John Wiley and Sons, New York, 1962.

[27] Hindmarsh, A. C., "GEAR: Ordinary Differential Equation System Solver," Report UCID-30001, Lawrence Livermore Laboratory, University of California, 1972.

[28] Hull, T. E., "The Numerical Integration of Ordinary Differential Equations," Information Processing, 68, ed. A. J. H. Morrell, North Holland Publishing Company, Amsterdam, pp. 40-53, 1969.

[29] Hull, T. E., and Enright, W. H., "A Structure for Programs That Solve Ordinary Differential Equations," Dept. of Computer Science, University of Toronto, Report #66, 1974.

[30] Jeltsch, R., "Multistep Methods Using Higher Derivatives and ε-Stability," submitted to SIAM J. Num. An.

[31] Lambert, J. D., and Sigurdsson, S. T., "Multistep Methods with Variable Matrix Coefficients," SIAM J. Num. An., 9, pp. 715-733, 1972.

[32] Liniger, W., and Willoughby, R., "Efficient Integration Method for Stiff Systems of Ordinary Differential Equations," SIGNUM, 7, No. 1, pp. 47-66, 1970.

[33] Moler, C. B., "Matrix Computations with FORTRAN and Paging," Comm. ACM, 15, pp. 268-271, 1972.

[34] Moulton, F. R., New Methods in Exterior Ballistics, University of Chicago, Chicago, Illinois, 1926.

[35] Nordsieck. A., "On the Numerical Integration of Ordinary Differential Equations," Math. Comp. 16, pp. 22-49, 1962.

[36] Norman, A. C., "TAYLOR User's Manual," University of Cambridge, Computer Laboratory, Cambridge, 1973.

[37] Ralston, A., A First Course in Numerical Analysis, McGraw-Hill, New York, 1965.

[38] Shampine, L. F., and Gordon, M. K., "Local Error and Variable Order, Variable Step Adams Codes," 1973.

[39] Skeel, R. D., "Convergence of Multivalue Methods for Solving Ordinary Differential Equations," Dept. of Computing Science, University of Alberta, Report TR73-16, 1973.

[40] Varga, R. S., "Some Results in Approximation Theory with Applications to Numerical Analysis," Numerical Solution of Partial Differential Equations: II, ed. B. E. Hubbard, Academic Press, New York, pp. 623-649, 1971.

[41] Widlund, O., "A Note on Unconditionally Stable Linear Multistep Methods," BIT, 1, pp. 65-70, 1967.

APPENDIX I

D4 FOR THE INTEGRATION OF STIFF SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS

```
      SUBROUTINE D4(N,T,Y,SAVE,H,HMIN,HMAX,DYS,DY,                      D4 001
     + EPS,YMAX,ERSV,SIGMA,PW,IP,KFLAG,JSTART,NQMAX)                    D4 002
      IMPLICIT REAL*8(A-H,O-Z)                                          D4 003
C*********************************************************************** D4 004
C*                                                                     * D4 005
C* THIS SUBROUTINE INTEGRATES A SET OF N ORDINARY DIFFERENTIAL FIRST   * D4 006
C* ORDER EQUATIONS OVER ONE STEP OF LENGTH H AT EACH CALL.  H CAN BE   * D4 007
C* SPECIFIED BY THE USER FOR EACH STEP, BUT IT MAY BE INCREASED OR     * D4 008
C* DECREASED BY D4 WITHIN THE RANGE HMIN TO HMAX IN ORDER TO           * D4 009
C* ACHIEVE AS LARGE A STEP AS POSSIBLE WHILE NOT COMMITTING A SINGLE   * D4 010
C* STEP ERROR WHICH IS LARGER THAN EPS IN THE L-2 NORM, WHERE EACH     * D4 011
C* COMPONENT OF THE ERROR IS DIVIDED BY THE COMPONENTS OF YMAX.        * D4 012
C*                                                                     * D4 013
C* THE PROGRAM REQUIRES FOUR SUBROUTINES NAMED                         * D4 014
C*    DIFFUN(T,H,Y,DYS,NORD)                                           * D4 015
C*    DECOMP(N,N,PW,IP)                                                * D4 016
C*    SOLVE(N,N,PW,DY,IP)                                              * D4 017
C*    MATRIX(PW,T,N,Y,EPS,H,B,DYS,NORD)                                * D4 018
C* THE FIRST, DIFFUN, EVALUATES THE DERIVATIVES OF THE DEPENDENT       * D4 019
C* VARIABLES STORED IN Y(1,I) FOR I = 1 TO N, AND STORES THE           * D4 020
C* DERIVATIVES IN THE ARRAY DYS.   DECOMP IS A                         * D4 021
C* STANDARD LU DECOMPOSITION WITH PIVOTING THAT DECOMPOSES THE MATRIX  * D4 022
C* PW, LEAVING THE PIVOTS IN THE INTEGER ARRAY IP.  N IS THE DECLARED  * D4 023
C* SIZE OF PW.  IP(N) IS SET TO 0 IF PW IS SINGULAR.  SOLVE PERFORMS   * D4 024
C* BACK SUBSTITUTION ON THE CONTENTS OF DY(I), LEAVING THE             * D4 025
C* RESULTS THERE.  THE MATRIX IN BOTH SHOULD BE SINGLE PRECISION,      * D4 026
C* WHILE ALL FLOATING POINT VECTORS SHOULD BE DOUBLE PRECISION.        * D4 027
C* MATRIX COMPUTES THE PARTIALS OF THE FIRST NORD DERIVATIVES AND      * D4 028
C* FORMS THE MATRIX PW.  A VERSION OF MATRIX USING NUMERICAL DIFFER-   * D4 029
C* ENCING FOLLOWS THIS LISTING.                                        * D4 030
C*                                                                     * D4 031
C* THE PROGRAM USES DOUBLE PRECISION ARITHMETIC FOR ALL FLOATING       * D4 032
C* POINT VARIABLES EXCEPT THOSE STARTING WITH P.  THE FORMER ARE       * D4 033
C* SINGLE PRECISION TO SAVE TIME AND SPACE.                            * D4 034
C*                                                                     * D4 035
C* THE TEMPORARY STORAGE SPACE IS PROVIDED BY THE CALLER IN THE        * D4 036
C* INTEGER ARRAY IP, THE SINGLE PREC. ARRAY PW, AND THE DOUBLE PREC.   * D4 037
C* ARRAYS SAVE,DYS,DY,SIGMA, AND ERSV.  PW IS USED ONLY TO HOLD        * D4 038
C* THE MATRIX OF THE SAME NAME, AND SAVE IS USED TO SAVE THE VALUES    * D4 039
C* OF Y IN CASE A STEP HAS TO BE REPEATED.                             * D4 040
C*                                                                     * D4 041
C* THE PARAMETERS TO THE SUBROUTINE D4 HAVE                            * D4 042
C* THE FOLLOWING MEANINGS..                                            * D4 043
C*                                                                     * D4 044
C*   N       THE NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS.         * D4 045
C*   T       THE INDEPENDENT VARIABLE.                                 * D4 046
C*   Y       A 12 BY N ARRAY CONTAINING THE DEPENDENT VARIABLES AND    * D4 047
C*             THEIR SCALED DERIVATIVES.  Y(J+1,I) CONTAINS            * D4 048
C*             THE J-TH DERIVATIVE OF Y(I) SCALED BY                   * D4 049
C*             H**J/FACTORIAL(J) WHERE H IS THE CURRENT                * D4 050
C*             STEP SIZE. ONLY Y(1,I) NEED BE PROVIDED BY              * D4 051
C*             THE CALLING PROGRAM ON THE FIRST ENTRY.                 * D4 052
C*               IF IT IS DESIRED TO INTERPOLATE TO NON MESH POINTS    * D4 053
C*             THESE VALUES CAN BE USED.  IF THE CURRENT STEP SIZE     * D4 054
C*             IS H AND THE VALUE AT T + E IS NEEDED, FORM             * D4 055
C*             S = E/H, AND THEN COMPUTE                               * D4 056
C*                                NQ                                   * D4 057
C*               Y(I)(T+E) =  SUM  Y(J+1,I)*S**J                       * D4 058
C*                            J=0                                      * D4 059
C*   SAVE    A BLOCK OF AT LEAST 12*N FLOATING POINT LOCATIONS.        * D4 060
C*   DY      AN ARRAY OF N DOUBLE PRECISION LOCATIONS.                 * D4 061
```

```
C*    ERSV     AN ARRAY OF N DOUBLE PRECISION LOCATIONS.                  *  D4 062
C*    SIGMA    AN ARRAY OF N DOUBLE PRECISION LOCATIONS.                  *  D4 063
C*    DYS      AN N BY 8 DIMENSIONED ARRAY THAT HOLDS THE JTH             *  D4 064
C*               DERIVATIVE ELEMENT OF THE TAYLOR SERIES FOR Y(I) IN      *  D4 065
C*               DYS(I,J).                                                *  D4 066
C*    H        THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP.            *  D4 067
C*               H MAY BE ADJUSTED UP OR DOWN BY THE PROGRAM              *  D4 068
C*               IN ORDER TO ACHIEVE AN ECONOMICAL INTEGRATION.           *  D4 069
C*               HOWEVER, IF THE H PROVIDED BY THE USER DOES              *  D4 070
C*               NOT CAUSE A LARGER ERROR THAN REQUESTED, IT              *  D4 071
C*               WILL BE USED.  TO SAVE COMPUTER TIME, THE USER IS        *  D4 072
C*               ADVISED TO USE A FAIRLY SMALL STEP FOR THE FIRST         *  D4 073
C*               CALL.  IT WILL BE AUTOMATICALLY INCREASED LATER.         *  D4 074
C*    HMIN     THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE            *  D4 075
C*               INTEGRATION.  NOTE THAT ON STARTING THIS MUST BE         *  D4 076
C*               MUCH SMALLER THAN THE AVERAGE H EXPECTED SINCE           *  D4 077
C*               A FIRST ORDER METHOD IS USED INITIALLY.                  *  D4 078
C*    HMAX     THE MAXIMUM SIZE TO WHICH THE STEP WILL BE INCREASED       *  D4 079
C*    EPS      THE ERROR TEST CONSTANT.  SINGLE STEP ERROR ESTIMATES      *  D4 080
C*               DIVIDED BY  YMAX(I)  MUST BE LESS THAN THIS              *  D4 081
C*               IN THE EUCLIDEAN NORM. THE STEP AND/OR ORDER IS          *  D4 082
C*               ADJUSTED TO ACHIEVE THIS.                                *  D4 083
C*    YMAX     AN ARRAY OF  N LOCATIONS WHICH CONTAINS THE MAXIMUM        *  D4 084
C*               OF EACH Y SEEN SO FAR.  IT SHOULD NORMALLY BE SET TO     *  D4 085
C*               1 IN EACH COMPONENT BEFORE THE FIRST ENTRY. (SEE THE     *  D4 086
C*               DESCRIPTION OF EPS.)                                     *  D4 087
C*    KFLAG    A COMPLETION CODE WITH THE FOLLOWING MEANINGS..            *  D4 088
C*                      +1  THE STEP WAS SUCCESFUL.                       *  D4 089
C*                      -1  THE STEP WAS TAKEN WITH H = HMIN, BUT THE     *  D4 090
C*                            REQUESTED ERROR WAS NOT ACHIEVED.           *  D4 091
C*                      -2  CORRECTOR CONVERGENCE COULD NOT BE ACHIEVED   *  D4 092
C*                            FOR H .GT. HMIN.                            *  D4 093
C*                      -3  PW IS NEARLY SINGULAR.                        *  D4 094
C*    JSTART   AN INPUT INDICATOR WITH THE FOLLOWING MEANINGS..           *  D4 095
C*                       0  PERFORM THE FIRST STEP.  THE FIRST STEP       *  D4 096
C*                            MUST BE DONE WITH THIS VALUE OF JSTART      *  D4 097
C*                            SO THAT THE SUBROUTINE CAN INITIALIZE       *  D4 098
C*                            ITSELF.                                     *  D4 099
C*                      +1  TAKE A NEW STEP CONTINUING FROM THE LAST.     *  D4 100
C*               JSTART IS SET TO NQ, THE CURRENT ORDER OF THE METHOD     *  D4 101
C*               AT EXIT.  NQ IS ALSO THE ORDER OF THE MAXIMUM            *  D4 102
C*               DERIVATIVE AVAILABLE.                                    *  D4 103
C*    NQMAX    THE MAXIMUM DERIVATIVE THAT SHOULD BE STORED IN THE        *  D4 104
C*               ARRAY Y.  SINCE THE ORDER IS EQUAL TO THE HIGHEST        *  D4 105
C*               DERIVATIVE USED, THIS RESTRICTS THE ORDER. IT MUST       *  D4 106
C*               BE LESS THAN 12.                                         *  D4 107
C*    PW       A BLOCK OF AT LEAST N**2 FLOATING POINT LOCATIONS.         *  D4 108
C*************************************************************************  D4 109
      DIMENSION Y(12,1),YMAX(1),SAVE(12,1),PW(N,N),IP(1),                   D4 110
     + ERSV(1),DY(1),SIGMA(1),DYS(N,1),                                     D4 111
     + B(4),A(12,4),EC(7)                                                   D4 112
      EQUIVALENCE (EC(1),E),(EC(2),EUP),(EC(3),EDWN),(EC(4),ENQ1),          D4 113
     + (EC(5),ENQ2),(EC(6),ENQ3),(EC(7),BND)                                D4 114
      IF(JSTART.GT.0)GO TO 100                                              D4 115
C*************************************************************************  D4 116
C* ON THE FIRST CALL, THE ORDER IS SET TO 1 AND THE INITIAL             *  D4 117
C* DERIVATIVES ARE CALCULATED.                                          *  D4 118
C*************************************************************************  D4 119
      K=2                                                                   D4 120
      NQ=1                                                                  D4 121
      NORD=1                                                                D4 122
```

```
      ID=2                                                              D4 123
      CALL DIFFUN(T,H,Y,DYS,NORD)                                       D4 124
      IWEVAL=1                                                          D4 125
      DO 10 I=1,N                                                       D4 126
   10 Y(2,I)=DYS(I,1)                                                   D4 127
      HNEW=H                                                            D4 128
C*********************************************************************  D4 129
C* BEGIN BY SAVING INFORMATION FOR POSSIBLE RESTARTS AND CHANGING    *  D4 130
C* H BY THE FACTOR H/HNEW IF THE CALLER HAS CHANGED H.  ALL VARIABLES *  D4 131
C* DEPENDENT ON H MUST ALSO BE CHANGED.                              *  D4 132
C* E IS A COMPARISON FOR ERRORS OF THE CURRENT ORDER NQ. EUP IS       *  D4 133
C* TO TEST FOR INCREASING THE ORDER, EDWN FOR DECREASING THE ORDER.   *  D4 134
C* HNEW IS THE STEP SIZE THAT WAS USED ON THE LAST CALL.             *  D4 135
C*********************************************************************  D4 136
  100 IF(H.NE.HNEW)CALL SCALE(Y,Y,H/HNEW,K,N)                           D4 137
      KFLAG=1                                                           D4 138
      DO 110 I=1,N                                                      D4 139
      DO 110 J=1,K                                                      D4 140
  110 SAVE(J,I)=Y(J,I)                                                  D4 141
      HOLD=H                                                            D4 142
      TOLD=T                                                            D4 143
      IF(JSTART.GT.0)GO TO 200                                          D4 144
C*********************************************************************  D4 145
C* CALL COEF.   THIS WILL  .  .  .                                   *  D4 146
C* SET THE COEFFICIENTS THAT DETERMINE THE ORDER AND THE METHOD      *  D4 147
C* TYPE.  CHECK FOR EXCESSIVE ORDER.  THE LAST TWO STATEMENTS OF     *  D4 148
C* THIS SECTION  SET IWEVAL .GT.0 IF PW IS TO BE RE-EVALUATED        *  D4 149
C*********************************************************************  D4 150
  150 CALL COEF(A,B,NORD,NQ,K,EC,IWEVAL,EPS)                            D4 151
C*********************************************************************  D4 152
C* THIS SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY          *  D4 153
C* MULTIPLYING THE SAVED INFORMATION BY THE PASCAL TRIANGLE           *  D4 154
C* MATRIX.                                                           *  D4 155
C*********************************************************************  D4 156
  200 T=T+H                                                             D4 157
      DO 210 J=2,K                                                      D4 158
      DO 210 J1=J,K                                                     D4 159
      J2=K-J1+J-1                                                       D4 160
      DO 210 I=1,N                                                      D4 161
  210 Y(J2,I)=Y(J2,I)+Y(J2+1,I)                                         D4 162
C* PERFORM UP TO 3 CORRECTOR ITERATIONS.  CALCULATE SIGMA(I), THE    *  D4 163
C* LINEAR COMBINATION OF PREVIOUS VALUES Y(I) AT T-J*H, J=1,2,...., ON *  D4 164
C* THE FIRST PASS.  IF THE CORRECTOR DOES NOT CONVERGE, GO TO 300,   *  D4 165
C* REDUCE H, AND SET IWEVAL=1 TO CAUSE PW TO BE REEVALUATED.  IF     *  D4 166
C* SUCCESSFUL, GO TO 350.                                            *  D4 167
      DO 300 L=1,3                                                      D4 168
  230 NT=N                                                              D4 169
      IF(L.GT.1)GO TO 250                                               D4 170
      DO 240 I=1,N                                                      D4 171
      SIGMA(I)=Y(1,I)                                                   D4 172
      DO 240 J=1,NORD                                                   D4 173
  240 SIGMA(I)=SIGMA(I)-B(J)*Y(J+1,I)                                   D4 174
  250 CALL DIFFUN(T,H,Y,DYS,NORD)                                       D4 175
      IF(IWEVAL.LE.0)GO TO 255                                          D4 176
      CALL MATRIX(PW,T,N,Y,EPS,H,B,DYS,NORD)                            D4 177
      CALL DECOMP(N,N,PW,IP)                                            D4 178
      IF(IP(N).EQ.0)GO TO 640                                           D4 179
      IWEVAL=-1                                                         D4 180
  255 CONTINUE                                                          D4 181
      DO 260 I=1,N                                                      D4 182
      DY(I)=SIGMA(I)-Y(1,I)                                             D4 183
```

```
      DO 260 J=1,NORD                                                    D4 184
  260 DY(I)=DY(I)+DYS(I,J)*B(J)                                          D4 185
      CALL SOLVE(N,N,PW,DY,IP)                                           D4 186
      DO 270 I=1,N                                                       D4 187
      Y(1,I)=Y(1,I)+DY(I)                                                D4 188
  270 IF(DABS(DY(I))/YMAX(I).LE.BND)NT=NT-1                              D4 189
      IF(NT.GT.0)GO TO 300                                               D4 190
      DO 280 I=1,N                                                       D4 191
      DO 280 J=1,NORD                                                    D4 192
      XK=DYS(I,J)                                                        D4 193
      DYS(I,J)=XK-Y(J+1,I)                                               D4 194
  280 Y(J+1,I)=XK                                                        D4 195
      GO TO 350                                                          D4 196
  300 CONTINUE                                                           D4 197
      H=DMAX1(HMIN,.2500*H)                                              D4 198
      IF(H.LT.1.100*HMIN)GO TO 630                                       D4 199
  301 T=TOLD                                                             D4 200
      ID=K                                                               D4 201
      IWEVAL=1                                                           D4 202
      R=H/HOLD                                                           D4 203
      CALL SCALE(Y,SAVE,R,K,N)                                           D4 204
      GO TO 200                                                          D4 205
  350 D=0.                                                               D4 206
      IF(NQ.EQ.1)GO TO 361                                               D4 207
  355 DO 360 I=1,N                                                       D4 208
      DD=0.D0                                                            D4 209
      DO 358 J=1,NORD                                                    D4 210
  358 DD=DD+DYS(I,J)*A(K,J)                                              D4 211
  360 D=D+(DD/YMAX(I))**2                                                D4 212
      GO TO 365                                                          D4 213
  361 DO 362 I=1,N                                                       D4 214
  362 D=D+((Y(2,I)-SAVE(2,I)*H/HOLD)/YMAX(I))**2                         D4 215
  365 IWEVAL=0                                                           D4 216
      IF(D.GT.E.AND.H.GT.1.1D0*HMIN)GO TO 450                            D4 217
C*********************************************************************  D4 218
C* THE CORRECTOR CONVERGED AND CONTROL IS PASSED TO STATEMENT 400     *  D4 219
C*  IF THE ERROR TEST IS O.K., AND TO 450 OTHERWISE.                  *  D4 220
C* IF THE STEP IS O.K.  IT IS ACCEPTED.  IF ID HAS BEEN REDUCED       *  D4 221
C* TO ZERO, A TEST IS MADE TO SEE IF THE STEP CAN BE INCREASED        *  D4 222
C* AT THE CURRENT ORDER OR BY GOING TO ONE HIGHER OR ONE LOWER.       *  D4 223
C* SUCH A CHANGE IS ONLY MADE IF THE STEP CAN BE INCREASED BY AT      *  D4 224
C* LEAST 1.1.  IF NO CHANGE IS POSSIBLE ID IS SET TO 7 TO             *  D4 225
C* PREVENT FUTHER TESTING FOR 8 STEPS.                               *  D4 226
C* IF A CHANGE IS POSSIBLE, IT IS MADE AND IDOUB IS SET TO            *  D4 227
C* NQ + 1  TO PREVENT FURTHER TESTING FOR K+1 STEPS.                 *  D4 228
C* IF THE ERROR WAS TOO LARGE, THE OPTIMUM STEP SIZE FOR THIS OR      *  D4 229
C* LOWER ORDER IS COMPUTED, AND THE STEP RETRIED.  IF IT SHOULD       *  D4 230
C* FAIL TWICE MORE IT IS AN INDICATION THAT THE DERIVATIVES THAT      *  D4 231
C* HAVE ACCUMULATED IN THE Y ARRAY HAVE ERRORS OF THE WRONG ORDER     *  D4 232
C* SO THE FIRST DERIVATIVES ARE RECOMPUTED AND THE ORDER IS SET       *  D4 233
C* TO 1.                                                             *  D4 234
C*********************************************************************  D4 235
C*********************************************************************  D4 236
C* COMPLETE THE CORRECTION OF THE HIGHER ORDER DERIVATIVES AFTER A    *  D4 237
C* SUCCESFUL STEP.                                                   *  D4 238
C*********************************************************************  D4 239
      IF(K.EQ.NORD+1)GO TO 400                                           D4 240
      ND=NORD+2                                                          D4 241
      DO 370 L=1,NORD                                                    D4 242
      DO 370 J=ND,K                                                      D4 243
      DO 370 I=1,N                                                       D4 244
```

```
  370 Y(J,I)=Y(J,I)+A(J,L)*DYS(I,L)                                    D4 245
  400 KFLAG=1                                                          D4 246
      ID=ID-1                                                          D4 247
      IF(ID.NE.1)GO TO 390                                             D4 248
       DO 380 I=1,N                                                    D4 249
  380 ERSV(I)=DYS(I,NORD)                                              D4 250
  390 IF(ID.GE.1)GO TO 575                                             D4 251
  395 PR2=(D/E)**ENQ2*1.2                                              D4 252
      PR3=1.E+20                                                       D4 253
      IF(KFLAG.LE.-1.OR.K.GT.NQMAX)GO TO 420                           D4 254
      D=0.                                                             D4 255
      A(2,1)=1.D0                                                      D4 256
      DO 410 I=1,N                                                     D4 257
      DD=A(K,NORD)*(DYS(I,NORD)-ERSV(I))                               D4 258
      IF(NORD.EQ.1)GO TO 410                                           D4 259
      DO 405 L=2,NORD                                                  D4 260
  405 DD=DD+A(K,L-1)*DYS(I,L)*DFLOAT(L)                                D4 261
  410 D=D+(DD/YMAX(I))**2                                              D4 262
      PR3=(D/EUP)**ENQ3*1.3                                            D4 263
  420 PR1=1.E+20                                                       D4 264
      D=0.                                                             D4 265
      IF(K.EQ.2) GO TO 460                                             D4 266
      DO 430 I=1,N                                                     D4 267
  430 D=D+(Y(K,I)/YMAX(I))**2                                          D4 268
      PR1=(D/EDWN)**ENQ1*1.3                                           D4 269
      GO TO 460                                                        D4 270
C*********************************************************************D4 271
C* REDUCE THE FAILURE FLAG COUNT TO CHECK FOR MULTIPLE FAILURES.     *D4 272
C* RESTORE T TO ITS ORIGINAL VALUE AND TRY AGAIN UNLESS THERE HAVE BEEN*D4 273
C* THREE FAILURES.  IN THAT CASE THE DERIVATIVES ARE ASSUMED TO HAVE  *D4 274
C* ACCUMULATED ERRORS SO A RESTART FROM THE CURRENT VALUES OF Y IS    *D4 275
C* TRIED. THIS IS CONTINUED UNTIL SUCCESS OR H = HMIN.               *D4 276
C* PR1, PR2, AND PR3  WILL CONTAIN THE AMOUNTS BY WHICH THE STEP SIZE *D4 277
C* SHOULD BE DIVIDED AT ORDER ONE LOWER, AT THIS ORDER, AND AT ORDER  *D4 278
C* ONE HIGHER RESPECTIVELY.                                          *D4 279
C*********************************************************************D4 280
  450 KFLAG=KFLAG-2                                                    D4 281
      IF(H.LT.1.1D0*HMIN)GO TO 620                                     D4 282
      T=TOLD                                                           D4 283
      IF(KFLAG.LE.-5)GO TO 600                                         D4 284
      GO TO 395                                                        D4 285
  460 IF(PR2.LE.PR3)GO TO 510                                          D4 286
      IF(PR3.LT.PR1)GO TO 520                                          D4 287
  470 R=1./AMAX1(PR1,1.E-4)                                            D4 288
      NEWQ=NQ-1                                                        D4 289
  480 IF(KFLAG.EQ.1.AND.R.LT.1.1)GO TO 575                             D4 290
      IF(NEWQ.LE.NQ)GO TO 500                                          D4 291
      IF(NQ.EQ.1)GO TO 495                                             D4 292
      DO 490 I=1,N                                                     D4 293
      Y(NEWQ+1,I)=0.D0                                                 D4 294
      DO 485 L=1,NORD                                                  D4 295
  485 Y(NEWQ+1,I)=Y(NEWQ+1,I)+DYS(I,L)*A(K,L)                          D4 296
  490 Y(NEWQ+1,I)=Y(NEWQ+1,I)/DFLOAT(NEWQ)                             D4 297
      GO TO 500                                                        D4 298
  495 DO 496 I=1,N                                                     D4 299
  496 Y(3,I)=(Y(2,I)-SAVE(2,I))/2.D0                                   D4 300
  500 K=NEWQ+1                                                         D4 301
      ID=K                                                             D4 302
      NQOLD=NQ                                                         D4 303
      NQ=NEWQ                                                          D4 304
      HNEW=DMAX1(HMIN,DMIN1(HMAX,H*R))                                 D4 305
```

```
      IF(KFLAG.EQ.1)GO TO 570                                          D4 306
      R=HNEW/HOLD                                                      D4 307
      H=HNEW                                                           D4 308
      IWEVAL=1                                                         D4 309
      CALL SCALE(Y,SAVE,R,K,N)                                         D4 310
      IF(NEWQ.EQ.NQOLD)GO TO 200                                       D4 311
      GO TO 150                                                        D4 312
  510 IF(PR2.GT.PR1)GO TO 470                                          D4 313
      NEWQ=NQ                                                          D4 314
      R=1./AMAX1(PR2,1.E-4)                                            D4 315
       GO TO 480                                                       D4 316
  520 R=1./AMAX1(PR3,1.E-4)                                            D4 317
      NEWQ=NQ+1                                                        D4 318
      GO TO 480                                                        D4 319
  570 R=HNEW/H                                                         D4 320
      H=HNEW                                                           D4 321
      CALL COEF(A,B,NORD,NQ,K,EC,IWEVAL,EPS)                           D4 322
      CALL SCALE(Y,Y,R,K,N)                                            D4 323
  575 DO 580 I=1,N                                                     D4 324
  580 YMAX(I)=DMAX1(YMAX(I),DABS(Y(1,I)))                              D4 325
  585 JSTART=NQ                                                        D4 326
      HNEW=H                                                           D4 327
      RETURN                                                           D4 328
  600 NQ=1                                                             D4 329
      K=2                                                              D4 330
      R=H/HOLD                                                         D4 331
      DO 610 I=1,N                                                     D4 332
      Y(1,I)=SAVE(1,I)                                                 D4 333
      Y(2,I)=SAVE(2,I)*R                                               D4 334
  610 CONTINUE                                                         D4 335
      KFLAG=1                                                          D4 336
      GO TO 150                                                        D4 337
  620 KFLAG=-1                                                         D4 338
      GO TO 650                                                        D4 339
  630 KFLAG=-2                                                         D4 340
      GO TO 650                                                        D4 341
  640 KFLAG=-3                                                         D4 342
  650 CALL SCALE(Y,SAVE,1.D0,K,N)                                      D4 343
      H=HOLD                                                           D4 344
      T=TOLD                                                           D4 345
      GO TO 585                                                        D4 346
      END                                                              D4 347
```

```
      SUBROUTINE COEF(A,B,NORD,NQ,K,EC,IWEVAL,EPS)                     D4 348
C*********************************************************************** D4 349
C* THIS SUBROUTINE PLACES THE CORRECTOR COEFFICIENTS FOR THE NORDSIECK * D4 350
C* FORM OF THE MULTIDERIVATIVE METHOD OF D4 INTO A AND B.              * D4 351
C*                                                                     * D4 352
C*   A(I,J)  THE COEFFICIENT OF DYS(L,J) TO UPDATE Y(I,L).             * D4 353
C*   B(J)    THE COEFFICIENT OF DYS(L,J) USED TO UPDATE Y(1,L).        * D4 354
C*   NORD    THE CURRENT HIGHEST DERIVATIVE BEING EVALUATED.           * D4 355
C*   NQ      THE CURRENT METHOD ORDER.                                 * D4 356
C*   K       SET TO NQ+1.                                              * D4 357
C*   IWEVAL  SET TO 1.                                                 * D4 358
C*   EPS     THE ERROR PARAMETER.                                      * D4 359
C*   EC      PARAMETERS USED IN THE COMPUTATION OF THE NEXT STEPSIZE   * D4 360
C*             AT ORDERS NQ, NQ-1, NQ+1.  SEE THE EQUIVALENCE STATEMENT* D4 361
C*             IN D4 FOR DETAILS.                                      * D4 362
C*                                                                     * D4 363
C* THE RATIONAL VALUES OF THE B(J) AND A(I,J) PRECEDE THE APPROPRIATE  * D4 364
C* SET OF ASSIGNMENT STATEMENTS FOR UP TO ORDER 8 METHODS.  THEY ARE   * D4 365
C* GIVEN IN THE SAME ORDER AS THE ASSIGNMENT STATEMENTS.  A(I+1,I) IS  * D4 366
```

```
C* ONE, A(1,J)=B(J)/FACTORIAL(J), AND OTHER A(I,J) NOT GIVEN ARE ZERO. *  D4 367
C*************************************************************************  D4 368
      IMPLICIT REAL*8 (A-H,O-Z)                                            D4 369
      DIMENSION PFAC(11),PC(12),A(12,4),EC(7),B(4)                         D4 370
      DATA PFAC/1.,2.,6.,24.,120.,720.,5040.,40320.,362880.,3628800.,      D4 371
     + 39916800./                                                          D4 372
      DATA PC/1.,.5,.3333333,.05555555,.027272727,                         D4 373
     +      .016,.00165232358,.891445687185E-3,                           D4 374
     +      .816463697E-4,.42443614719E-4,.239909408E-4,                  D4 375
     +      .144629115175E-4/                                             D4 376
      NO=MINO(NQ,10)                                                       D4 377
      GO TO(150,151,152,153,154,155,156,157,158,159,160),NQ                D4 378
  150 NORD=1                                                               D4 379
C**********                                                               D4 380
C* 1                                                                       D4 381
C**********                                                               D4 382
      B(1)=1.D0                                                            D4 383
      GO TO 190                                                            D4 384
  151 NORD=1                                                               D4 385
C**********                                                               D4 386
C* 2/3,1/3                                                                 D4 387
C**********                                                               D4 388
      B(1)=.6666666666666666D0                                             D4 389
      A(3,1)=.333333333333333D0                                            D4 390
      GO TO 190                                                            D4 391
  152 NORD=2                                                               D4 392
C**********                                                               D4 393
C* 6/7,-4/7,-1/7,3/7                                                       D4 394
C**********                                                               D4 395
      B(1)=.857142857142857142857D0                                        D4 396
      B(2)=-.571428571428571428D0                                          D4 397
      A(4,1)=-.142857142857142857D0                                        D4 398
      A(4,2)=.428571428571428571D0                                         D4 399
      GO TO 190                                                            D4 400
  153 NORD=2                                                               D4 401
C**********                                                               D4 402
C* 66/85,-36/85,-5/17,-6/85,12/17,11/85                                    D4 403
C**********                                                               D4 404
      B(1)=.7764705882352941D0                                             D4 405
      B(2)=-.4235294117647058D0                                            D4 406
      A(4,1)=-.2941176470588241D0                                          D4 407
      A(5,1)=-.07058823529411765D0                                         D4 408
      A(4,2)=.7058823529411765D0                                           D4 409
      A(5,2)=.1294117647058820D0                                           D4 410
      GO TO 190                                                            D4 411
  154 NORD = 2                                                             D4 412
C**********                                                               D4 413
C* 60/83,-144/415,-145/332,-15/83,-7/332,151/166,119/415,5/166             D4 414
C**********                                                               D4 415
      B(1)=.7228915662650601D0                                             D4 416
      B(2)=-.3469879518072291D0                                            D4 417
      A(4,1)=-.4367469879518101D0                                          D4 418
      A(5,1)=-.1807228915662701D0                                          D4 419
      A(6,1)=-.0210843373493980D0                                          D4 420
      A(4,2)=.9096385542168701D0                                           D4 421
      A(5,2)=.2867469879518101D0                                           D4 422
      A(6,2)=.0301204819277108D0                                           D4 423
      GO TO 190                                                            D4 424
C**********                                                               D4 425
C* -996/1169,720/1169,1728/5845,341/1336,41/334,145/9352,                  D4 426
C*     -415/668,-45/167,-151/4676,1667/1670,55/167,83/2338                 D4 427
```

```
C********                                                                D4 428
  155 B(1)=.65201026518392D0                                             D4 429
      NORD=3                                                             D4 430
      B(2)=-.61591103507271170                                           D4 431
      B(3)=.29563729683490160                                            D4 432
      A(5,1)=.25523952095808D0                                           D4 433
      A(6,1)=.12275449101796400                                          D4 434
      A(7,1)=.01550470487596200                                          D4 435
      A(5,2)=-.62125748502990                                            D4 436
      A(6,2)=-.26946107784430                                            D4 437
      A(7,2)=-.03229255774166D0                                          D4 438
      A(5,3)=.9982035928143700                                           D4 439
      A(6,3)=.32934131736527D0                                           D4 440
      A(7,3)=.03550042771599D0                                           D4 441
      GO TO 190                                                          D4 442
C********                                                                D4 443
C* 103020/124977,-493200/874853,216000/874853,356895/999832,             D4 444
C*     228543/999832,51225/999832,3905/999832,-406285/499916,            D4 445
C*       -234285/499916,-348337/3499412,-3675/499916,294225/249958,       D4 446
C*         128845/249958,172065/1749706,1717/249958                       D4 447
C**********                                                              D4 448
  156 B(1)=.82429848214500                                               D4 449
      NORD=3                                                             D4 450
      B(2)=-.50375185316848D0                                            D4 451
      B(3)=.24689862182560570                                            D4 452
      A(5,1)=.35695496843468700                                          D4 453
      A(6,1)=.22858140167548200                                          D4 454
      A(7,1)=.05123360724601700                                          D4 455
      A(8,1)=.00390565150233240                                          D4 456
      A(5,2)=-.81270653469782700                                         D4 457
      A(6,2)=-.46864873298714D0                                          D4 458
      A(7,2)=-.09954158012831810                                         D4 459
      A(8,2)=-.00735123500748120                                         D4 460
      A(5,3)=1.17709775242240                                            D4 461
      A(6,3)=.51546659838852900                                          D4 462
      A(7,3)=.09833937815838780                                          D4 463
      A(8,3)=.00686915401787500                                          D4 464
      GO TO 190                                                          D4 465
  157 NORD=4                                                             D4 466
C********                                                                D4 467
C* 52491180/58067611,4326840/58067611,2959200/58067611,12960000/580676  D4 468
C*    135270975/929081776,31785985/929081776,428418347/116135222000,     D4 469
C*      492548490/929081776,323874474/929081776,1038652097/11613522200,  D4 470
C*        2843995/464540888,16279665/232270444,12791866 5/232270444,     D4 471
C*          7600419/232270444,2059575/232270444,64038605/116135222,      D4 472
C*            12401655/116135222.                                        D4 473
C**********                                                              D4 474
      B(1)=.90396658474549500                                            D4 475
      B(2)=-.74513828371550990                                           D4 476
      B(3)=.50961283735264780                                            D4 477
      B(4)=-.22318810395D0                                               D4 478
      A(6,1)=-.21010653318422800                                         D4 479
      A(7,1)=-.14559641410941400                                         D4 480
      A(8,1)=-.03421225754405710                                         D4 481
      A(9,1)=-.00368896136436565D0                                       D4 482
      A(6,2)=.53014546482721100                                          D4 483
      A(7,2)=.34859630483163400                                          D4 484
      A(8,2)=.08943471921034010                                          D4 485
      A(9,2)=.00612216292142669D0                                        D4 486
      A(6,3)=-.93115448214325700                                         D4 487
      A(7,3)=-.55072898125602600                                         D4 488
```

```
      A(8,3)=-.11882880371986400                                        D4 489
      A(9,3)=-.00886714195974150                                        D4 490
      A(6,4)=1.22897276590216D0                                         D4 491
      A(7,4)=.55141415237490900                                         D4 492
      A(8,4)=.10678633739556300                                         D4 493
      A(9,4)=.00753305487287976D0                                       D4 494
      GO TO 190                                                         D4 495
  158 NORD=4                                                            D4 496
      B(1)=.88879254103287900                                           D4 497
      B(2)=-.71239449558903590                                          D4 498
      B(3)=.46581061984546120                                           D4 499
      B(4)=-.19012678361040D0                                           D4 500
      A(6,1)=-.28179007301499400                                        D4 501
      A(7,1)=-.23558395893838D0                                         D4 502
      A(8,1)=-.07511004846618390                                        D4 503
      A(9,1)=-.01067526766214590                                        D4 504
      A(10,1)=-.00056656408683862D0                                     D4 505
      A(6,2)=.68483004889120D0                                          D4 506
      A(7,2)=.54277876252715D0                                          D4 507
      A(8,2)=.16768730541330800                                         D4 508
      A(9,2)=.02335566549202220                                         D4 509
      A(10,2)=.00122257815818899D0                                      D4 510
      A(6,3)=-1.13380800757966D0                                        D4 511
      A(7,3)=-.81049190840915900                                        D4 512
      A(8,3)=-.23688658498525900                                        D4 513
      A(9,3)=-.03192084683629640                                        D4 514
      A(10,3)=-.00163547461881029D0                                     D4 515
      A(6,4)=1.38515740052960D0                                         D4 516
      A(7,4)=.74747969050837600                                         D4 517
      A(8,4)=.19589475124168500                                         D4 518
      A(9,4)=.02493367898090260                                         D4 519
      A(10,4)=.00123443408478220                                        D4 520
      GO TO 190                                                         D4 521
  159 NORD=4                                                            D4 522
      B(1)=.87547758101935600                                           D4 523
      B(2)=-.68491322375695190                                          D4 524
      B(3)=.43131650690545400                                           D4 525
      B(4)=-.16634796409574731D0                                        D4 526
      A(6,1)=-.35636313612076D0                                         D4 527
      A(7,1)=-.34119753667435700                                        D4 528
      A(8,1)=-.13288215203663600                                        D4 529
      A(9,1)=-.02595078018885320                                        D4 530
      A(10,1)=-.00252271436104220                                       D4 531
      A(11,1)=-.00009726667473136870                                    D4 532
      A(6,2)=.83874433902666D0                                          D4 533
      A(7,2)=.76075879675207200                                         D4 534
      A(8,2)=.28692543162002400                                         D4 535
      A(9,2)=.05488339908598780                                         D4 536
      A(10,2)=.00525995396550927D0                                      D4 537
      A(11,2)=.00020075253142186100                                     D4 538
      A(6,3)=-1.33127116750330D0                                        D4 539
      A(7,3)=-1.08409743743743700                                       D4 540
      A(8,3)=-.38655263455005800                                        D4 541
      A(9,3)=-.07149402270591460                                        D4 542
      A(10,3)=-.00670313298374840                                       D4 543
      A(11,3)=-.00025198180543352200                                    D4 544
      A(6,4)=1.51833537528456D0                                         D4 545
      A(7,4)=.93609206341187800                                         D4 546
      A(8,4)=.29906835614792900                                         D4 547
      A(9,4)=.05221379510565050                                         D4 548
      A(10,4)=.00472786885816340                                        D4 549
```

```
      A(11,4)=.00017370586925364200                                    D4 550
      GO TO 190                                                        D4 551
  180 NORD=4                                                           D4 552
      B(1)=.86363881973027600                                          D4 553
      B(2)=-.66138633445985800                                         D4 554
      B(3)=.40330253380768070                                          D4 555
      B(4)=-.14838989417356029D0                                       D4 556
      A(6,1)=-.43283540780964800                                       D4 557
      A(7,1)=-.46001101220361500                                       D4 558
      A(8,1)=-.2071813194922D0                                         D4 559
      A(9,1)=-.04991149463483060                                       D4 560
      A(10,1)=-.00673312473421530                                      D4 561
      A(11,1)=-.00048028569187979D0                                    D4 562
      A(12,1)=-.00001412076538077D0                                    D4 563
      A(6,2)=.99071586935629D0                                         D4 564
      A(7,2)=.99687396433708D0                                         D4 565
      A(8,2)=.43457840348858D0                                         D4 566
      A(9,2)=.10249995815617D0                                         D4 567
      A(10,2)=.01362720255892070                                       D4 568
      A(11,2)=.00096191713259789600                                    D4 569
      A(12,2)=.00002806186186186186D0                                  D4 570
      A(6,3)=-1.5122269372386700                                       D4 571
      A(7,3)=-1.3652448840080D0                                        D4 572
      A(8,3)=-.56236620384223300                                       D4 573
      A(9,3)=-.12819208371779600                                       D4 574
      A(10,3)=-.01666619584487390                                      D4 575
      A(11,3)=-.00115831686701513D0                                    D4 576
      A(12,3)=-.00003341386246626420                                   D4 577
      A(6,4)=1.6343352141625800                                        D4 578
      A(7,4)=1.1163187483711800                                        D4 579
      A(8,4)=.41171840209900                                           D4 580
      A(9,4)=.08855950517403680                                        D4 581
      A(10,4)=.01111458804538900                                       D4 582
      A(11,4)=.00075470265794556500                                    D4 583
      A(12,4)=.00002141961357734700                                    D4 584
  190 K=NQ+1                                                           D4 585
  196 EC(5)=.5/DFLOAT(K)                                               D4 586
      EC(6)=.5/DFLOAT(NQ+2)                                            D4 587
      EC(4)=.5/DFLOAT(NQ)                                              D4 588
      EC(1)=(EPS/(PC(K)*PFAC(NQ)))**2                                  D4 589
      IF(NQ.GE.11)GO TO 197                                            D4 590
      EC(2)=(EPS/(PC(K+1)*PFAC(NQ)))**2                                D4 591
  197 CONTINUE                                                         D4 592
      EC(3)=(EPS/(PC(NQ)*PFAC(NQ)))**2                                 D4 593
      EC(7)=(EPS*EC(6))                                                D4 594
      IWEVAL=1                                                         D4 595
  200 RETURN                                                           D4 596
      END                                                              D4 597
```

```
      SUBROUTINE SCALE(Y,YS,R,K,N)                                     D4 598
      IMPLICIT REAL*8 (A-H,O-Z)                                        D4 599
      DIMENSION Y(12,1),YS(12,1)                                       D4 600
      R1=1.D0                                                          D4 601
      DO 10 J=1,K                                                      D4 602
      DO 9 I=1,N                                                       D4 603
    9 Y(J,I)=YS(J,I)*R1                                                D4 604
   10 R1=R1*R                                                          D4 605
      RETURN                                                           D4 606
      END                                                              D4 607
```

```
      SUBROUTINE MATRIX(PW,T,N,Y,EPS,H,R,DYS,NORD)                          D4 608
C*************************************************************************  D4 609
C* THIS SUBROUTINE TAKES N CALLS TO DIFFUN TO FIND THE PARTIAL          *  D4 610
C* DERIVATIVE WITH RESPECT TO Y(I),I=1,N, OF THE JTH DERIVATIVE IN T    *  D4 611
C* OF EACH OF THE VARIABLES , FOR J=1,NORD.  THE CHANGE IN Y(I) FOR     *  D4 612
C* FORWARD DIFFERENCING IS R = MAX(EPS*Y(I),EPS*EPS).                   *  D4 613
C* THESE ARE USED TO CALCULATE THE MATRIX PW.                           *  D4 614
C*                                                                      *  D4 615
C*                  NORD                                                *  D4 616
C*    PW       I - SUM   H**J * BETA(J) * DF((J-1))/DY                  *  D4 617
C*                  J=1                                                 *  D4 618
C*                WHERE ((J)) IS THE JTH DERIVATIVE IN T.               *  D4 619
C*    T         THE INDEPENDENT VARIABLE.                               *  D4 620
C*    N         THE NUMBER OF EQUATIONS; THE ORDER OF PW.               *  D4 621
C*    Y         DEPENDENT VARIABLE ARRAY AS IN D4.                      *  D4 622
C*    EPS       ERROR CRITERION                                         *  D4 623
C*    H         STEPSIZE                                                *  D4 624
C*    BETA(I) COEFFICIENT USED IN D4 = B(I)/FACTORIAL(I).               *  D4 625
C*    DYS(I,J) THE JTH DERIVATIVE ELEMENT OF THE TAYLOR SERIES FOR Y(I) *  D4 626
C*    NORD      THE CURRENT HIGHEST DERIVATIVE BEING EVALUATED.         *  D4 627
C*************************************************************************  D4 628
      IMPLICIT REAL*8 (A-H,O-Z)                                             D4 629
      COMMON NFNS,NS                                                        D4 630
      DIMENSION PW(N,N),Y(12,1),B(4),DYS(N,1)                               D4 631
      NFNS=NFNS-N                                                           D4 632
      DO 5 I=1,N                                                            D4 633
      DO 5 J=1,N                                                            D4 634
      PW(I,J)=0.                                                            D4 635
      IF(I.EQ.J)PW(I,J)=1.                                                  D4 636
    5 CONTINUE                                                              D4 637
      DO 20 I=1,N                                                           D4 638
      R=-EPS*DMAX1(EPS,DABS(Y(1,I)))                                        D4 639
      F=Y(1,I)                                                              D4 640
      Y(1,I)=Y(1,I)-R                                                       D4 641
      CALL DIFFUN(T,H,Y,DYS(1,5),NORD)                                      D4 642
      DO 10 K=1,N                                                           D4 643
      DO 10 J=1,NORD                                                        D4 644
   10 PW(K,I)=PW(K,I) + B(J)*(DYS(K,J+4)-DYS(K,J))/R                        D4 645
   20 Y(1,I)=F                                                              D4 646
      RETURN                                                                D4 647
      END                                                                   D4 648
```

```
      SUBROUTINE DECOMP(N,NDIM,A,IP)                                    D4 649
      IMPLICIT REAL*8(B-H,O-Z)                                          D4 650
C                                                                       D4 651
C     MATRIX TRIANGULARIZATION BY GAUSSIAN ELIMINATION.                 D4 652
C                                                                       D4 653
C   INPUT...                                                            D4 654
C     N = ORDER OF MATRIX                                               D4 655
C     NDIM = DECLARED DIMENSION OF ARRAY A.                             D4 656
C     A = MATRIX TO BE TRIANGULARIZED. (FOR STIFF METHODS, A IS SINGLE  D4 657
C                 PRECISION;  ALL OTHER VARIABLE ARE DOUBLE PRECISION.) D4 658
C   OUTPUT...                                                           D4 659
C     A(I,J), I.LE.J  = UPPER TRIANGULAR FACTOR, U.                     D4 660
C     A(I,J),   I.GT.J   = MULTIPLIERS  = LOWER TRIANGULAR FACTOR, I-L. D4 661
C     IP(K),  K.LT.N = INDEX OF K-TH PIVOT ROW.                         D4 662
C     IP(N)  = (-1)**(NUMBER OF INTERCHANGES) OR 0.                     D4 663
C   USE 'SOLVE' TO OBTAIN SOLUTION OF LINEAR SYSTEM.                    D4 664
C   DETERM(A) = IP(N)*A(1,1)*A(2,2)*...*A(N,N).                         D4 665
C   IF IP(N) = 0, A IS SINGULAR, SOLVE WILL DIVIDE BY ZERO.             D4 666
C                                                                       D4 667
      DIMENSION A(NDIM,NDIM),IP(NDIM)                                   D4 668
      IP(N) = 1                                                         D4 669
      DO 6  K=1,N                                                       D4 670
      IF(K.EQ.N) GO TO 5                                                D4 671
      KP1 = K+1                                                         D4 672
      M = K                                                             D4 673
      DO 1  I=KP1,N                                                     D4 674
      IF(ABS(A(I,K)).GT.ABS(A(M,K)))    M=I                             D4 675
    1 CONTINUE                                                          D4 676
      IP(K) = M                                                         D4 677
      IF(M.NE.K)  IP(N) = -IP(N)                                        D4 678
      T = A(M,K)                                                        D4 679
      A(M,K) = A(K,K)                                                   D4 680
      A(K,K) = T                                                        D4 681
      IF(T.EQ.0) GO TO 5                                                D4 682
      DO 2  I=KP1,N                                                     D4 683
    2    A(I,K) = -A(I,K) / T                                           D4 684
      DO 4  J=KP1,N                                                     D4 685
      T = A(M,J)                                                        D4 686
      A(M,J) = A(K,J)                                                   D4 687
      A(K,J) = T                                                        D4 688
      IF(T.EQ.0.) GO TO 4                                               D4 689
      DO 3   I=KP1,N                                                    D4 690
    3    A(I,J) = A(I,J) + A(I,K)*T                                     D4 691
    4  CONTINUE                                                         D4 692
    5 IF(A(K,K).EQ.0.) IP(N) = 0                                        D4 693
    6 CONTINUE                                                          D4 694
      RETURN                                                            D4 695
      END                                                               D4 696
```

```
      SUBROUTINE SOLVE(N,NDIM,A,B,IP)                                   D4 697
      IMPLICIT REAL*8 (B-H,O-Z)                                         D4 698
C                                                                       D4 699
C   SOLUTION OF LINEAR SYSTEM, A*X = B.                                 D4 700
C                                                                       D4 701
C   INPUT...                                                            D4 702
C     N = ORDER OF MATRIX.                                              D4 703
C     NDIM = DECLARED DIMENSION OF ARRAY A.                             D4 704
C     A = TRIANGULARIZED MATRIX OBTAINED FROM 'DECOMP'.                 D4 705
C     B = RIGHT HAND SIDE VECTOR.                                       D4 706
C     IP = PIVOT VECTOR OBTAINED FROM 'DECOMP'.                         D4 707
C   OUTPUT...                                                           D4 708
C     B = SOLUTION VECTOR, X.                                           D4 709
      DIMENSION A(NDIM,NDIM), B(NDIM), IP(NDIM)                         D4 710
      IF(N.EQ.1) GO TO 9                                                D4 711
      NM1 = N-1                                                         D4 712
      DO 7  K=1,NM1                                                     D4 713
         KP1 = K + 1                                                    D4 714
         M = IP(K)                                                      D4 715
       T=B(M)                                                           D4 716
      B(M) = B(K)                                                       D4 717
      B(K) = T                                                          D4 718
      DO 7  I=KP1,N                                                     D4 719
    7 B(I) = B(I) + A(I,K)*T                                            D4 720
      DO 8   KB=1,NM1                                                   D4 721
      KM1 = N - KB                                                      D4 722
      K = KM1 + 1                                                       D4 723
      B(K) = B(K)/A(K,K)                                                D4 724
      T = -B(K)                                                         D4 725
      DO 8  I=1,KM1                                                     D4 726
    8 B(I) = B(I) + A(I,K)*T                                            D4 727
    9 B(1) = B(1)/A(1,1)                                                D4 728
      RETURN                                                            D4 729
      END                                                               D4 730
```

APPENDIX II

TEST PROBLEMS FOR COMPARISON OF D4 WITH DIFSUB

A. $$\underline{y}' = \begin{bmatrix} -1 & 95 \\ -1 & -97 \end{bmatrix} \underline{y},\ y(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad t_f = 10.$$

B. $$y_1' = (-1 + y_2^2)y_1 + (1 + y_2)y_2,\ y_1(0) = -1,$$

$$y_2' = -y_1 + (-19 + y_1(2 + y_1))y_2,\ y_2(0) = 1,$$

$$t_f = 16.$$

C. $$\underline{y}' = \begin{bmatrix} -10^4 & 10^3 & 0 & 0 \\ -10^3 & -10^4 & 0 & 0 \\ 0 & 0 & -50 & 10 \\ 0 & 0 & -10 & 50 \end{bmatrix} \underline{y},\ \underline{y}(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

$$t_f = 3.$$

D. $$\underline{y}' = \begin{bmatrix} 2-3Q & 4-4Q \\ 1.5Q-1.5 & 2Q-3 \end{bmatrix} \underline{y}, \quad \underline{y}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

$$Q = 10^5,\ t_f = 10.$$

E. $$\underline{y}' = \begin{bmatrix} -10^4 & 9990 & 0 & 0 \\ -9990 & -10^4 & 0 & 0 \\ 0 & 0 & -.09 & .1 \\ 0 & 0 & -.1 & -.09 \end{bmatrix} \underline{y},\ \underline{y}(0) = \begin{bmatrix} 50 \\ 50 \\ 1 \\ 1 \end{bmatrix},$$

$$t_f = 100.$$

F. $y_1' = 100(y_4 - y_2) - y_1 - 2(y_2 - y_4)^2$

$y_2' = y_1$, $\underline{y}(0) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, $t_f = 64.$

$y_3' = .01(100y_2 - y_4 + 2(y_2 - y_4)^2)$

$y_4' = y_3$

G. $\underline{y}' = \begin{bmatrix} -5 & -5 & -45.5 & 44.5 \\ -5 & -5 & -44.5 & 45.5 \\ 45.5 & 44.5 & -5 & 5 \\ -44.5 & -45.5 & 5 & -5 \end{bmatrix} \underline{y}$, $\underline{y}(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$,

$t_f = 25.$

H. $y_1' = 100(y_4 - y_2) - y_1$

$y_2' = y_1$, $\underline{y}(0) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$,

$y_3' = .01(100y_2 - 101y_4 - 10y_4^2)$

$y_4' = y_3$

$t_f = 64.$

I. $\underline{y}' = B\,\underline{y} + f(t)$, $y(0) = [1 \quad 0 \quad 1 \quad 0]^T$

$$B = \begin{bmatrix} -.1 & -.05 & 0 & 0 \\ .05 & -.1 & 0 & 0 \\ 0 & 0 & -100 & -100 \\ 0 & 0 & 100 & -100 \end{bmatrix}, \quad f(t) = \begin{bmatrix} 1 + .05t + 3t^2 + .05t^3 \\ -1 - .15t = 3t^2 - .15t^3 \\ 2t - 4t^3 \\ -2t - 200t^2 + 4t^3 + 200t^4 \end{bmatrix}$$

$t_f = 100.$

J. $\underline{y}' = UBU^T\,\underline{y} + U\underline{w}$, $y(0) = \begin{bmatrix} 0 \\ -2 \\ -1 \\ -1 \end{bmatrix}$, $U = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 11 & 11 & -1 \end{bmatrix}$

$$B = \begin{bmatrix} 10 & 10 & 0 & 0 \\ -10 & 10 & 0 & 0 \\ 0 & 0 & -1000 & 0 \\ 0 & 0 & 0 & -.001 \end{bmatrix}, \quad \underline{w} = \begin{bmatrix} \frac{t_1^2 - t_2^2}{2} \\ t_1 t_2 \\ t_3^3 \\ t_4^2 \end{bmatrix}$$

$\underline{t} = U\underline{y}$, $t_f = 100.$

K. $\underline{y}' = UBU^T \underline{y} + U\, f(t)$, $\underline{y}(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$,

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -100 & -900 \\ 0 & 0 & 900 & -100 \end{bmatrix}, \qquad f(t) = \begin{bmatrix} t^2 + 2t \\ t^2 - 2t \\ -800 + 1 \\ -1000t - 1 \end{bmatrix},$$

U as in J., $t_f = 25$.

L. $\underline{y}' = UBU^T \underline{y} + U\underline{w}$, $y(0) = \begin{bmatrix} 0 \\ -2 \\ -1 \\ -1 \end{bmatrix}$, $B = \begin{bmatrix} 10 & 100 & 0 & 0 \\ -100 & 10 & 0 & 0 \\ 0 & 0 & -100 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$,

U and $\underline{w}$ as in J., $t_f = 25$.

M. $\underline{y}' = A\underline{y} + f(t)$, $\underline{y}(0) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$, $A = \begin{bmatrix} -100 & -1000 & 0 & 0 \\ 1000 & -100 & 0 & 0 \\ 0 & 0 & -.02 & 0 \\ 0 & 0 & 0 & -.01 \end{bmatrix}$,

$$f(t) = \begin{bmatrix} 100t^3 + 1000t^3/(1+t) - 3e^{-.01t} \\ -1000t^3 + 100t^3/(1+t) + (3t^2+2t^3)/(1+t)^2 \\ .02t^3 + 3t^2 \\ .01t^2 + 2t \end{bmatrix},$$

$t_f = 100.$

VITA

Roy Leonard Brown, Jr. was born in Montgomery, Alabama, on August 30, 1949. He received his B.S. in Mathematics and Physics from the Tulane University of Louisiana in New Orleans in 1971. He entered the Graduate College of the University of Illinois in June, 1971, working as a research assistant under C. W. Gear and receiving his M.S. in Computer Science in February, 1973.

Mr. Brown is a member of the Association for Computing Machinery and its Special Interest Group on Numerical Mathematics, the American Mathematical Society, and an associate member of Sigma Xi, the Research Society of America.

Form AEC–427
(6/68)
AECM 3201

U.S. ATOMIC ENERGY COMMISSION
UNIVERSITY–TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO.
COO-2383-0012

2. TITLE
Multi-Derivative Numerical Methods for the Solution of Stiff Ordinary Differential Equations

3. TYPE OF DOCUMENT (Check one):

- [X] a. Scientific and technical report
- [ ] b. Conference paper not to be published in a journal:
  - Title of conference _____
  - Date of conference _____
  - Exact location of conference _____
  - Sponsoring organization _____
- [ ] c. Other (Specify) _____

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

- [X] a. AEC's normal announcement and distribution procedures may be followed.
- [ ] b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
- [ ] c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

C. W. Gear
Professor and Principal Investigator

Organization
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Signature

Date
December, 1974

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

- [ ] a. AEC patant clearance has been granted by responsible AEC patent group.
- [ ] b. Report has been sent to responsible AEC patent group for clearance.
- [ ] c. Patent clearance not required.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-74-672	2.	3. Recipient's Accession No.
4. Title and Subtitle MULTI-DERIVATIVE NUMERICAL METHODS FOR THE SOLUTION OF STIFF ORDINARY DIFFERENTIAL EQUATIONS			5. Report Date 6.
7. Author(s) ROY LEONARD BROWN, JR.			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaign Urbana, IL 61801			10. Project/Task/Work Unit No. 11. Contract/Grant No. AEC AT (11-1) 2383
12. Sponsoring Organization Name and Address US AEC Chicago Operations Office 9800 South Cass Avenue Argonne, IL 60439			13. Type of Report & Period Covered 14.

15. Supplementary Notes

16. Abstracts

Current research in the physical sciences and engineering has greatly enhanced the importance of computer programs for the numerical solution of systems of ordinary differential equations. Recent developments have concentrated on the solution of systems characterized as stiff for which the requirement of numerical stability can cause some integration methods to use a smaller stepsize than the control of the local truncation error requires.

The stability and convergence of linear multi-step multi-derivative formulas have been investigated since it has been shown that linear multi-step formulas of order greater than two cannot be A-stable, a useful attribute for formulas used to integrate stiff systems. It has been shown that when a stable and consistent linear k-step s-derivative formula implemented in an iterated corrector method is applied to a system satisfying certain continuity requirements, the solution is convergent.

A set of 11 multi-derivative multi-step formulas, all A-stable and of orders 1 through 11, have been found and implemented in a variable stepsize, variable order method as a FORTRAN program called D4. Tests of D4 on a set of stiff problems show it compares favorably with another stiff integrator.

17. Key Words and Document Analysis. 17a. Descriptors

Ordinary Differential Equations
Stiff Equations
Numerical Integration

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Unlimited	UNCLASSIFIED	99
	20. Security Class (This Page) UNCLASSIFIED	22. Price

MAR 7 - 1975

SEP 5 1975

UNIVERSITY OF ILLINOIS-URBANA
510.84 IL8R no. C002 no.867-872(1974
Report /
3 0112 088401382